Orchestration in a 5G-MEC Testbed for V2X Applications

by

Prachi Vinod Wadatkar

Thesis submitted in fulfillment of
the requirements for the degree of

PHILOSOPHIAE DOCTOR
(PhD)

Supervisors:

Assoc. Prof. Gianfranco Nencioni
Assoc. Prof. Rosario G. Garroppo

Faculty of Science and Technology
Department of Electrical
Engineering and Computer Science

Department of Information
Engineering

2024

University of Stavanger
N-4036 Stavanger
NORWAY
www.uis.no

ISBN 978-82-8439-276-9
ISSN 1890-1387

PhD Thesis UiS no. 790

Preface

This thesis is presented as a partial requirement for the attainment of the *Philosophiae Doctor* degree at the University of Stavanger, Norway, and the University of Pisa, Italy. The doctoral research activities have been supported by the *5G-MODaNeI* project, funded by the Norwegian Research Council (project no. 308909). The doctoral thesis is part of a cotutelle agreement between the University of Stavanger and the University of Pisa. The research was conducted within the Department of Electrical Engineering and Computer Science at the University of Stavanger from January 2021 to March 2024, excluding a research visit to the Department of Information Engineering at the University of Pisa from November 2022 to June 2023. Compulsory courses were completed at the University of Stavanger and the University of Pisa.

This thesis is comprised of a compilation of five published papers. Some tables have been divided into two parts to improve visibility and align with the page layout requirements of the thesis.

Prachi Vinod Wadatkar, August 2024

Abstract

The Fifth Generation (5G) of mobile networks is revolutionizing connectivity, enabling faster data transmission and lower latency. 5G enhances the performance of the usage scenarios supported by the previous generation but also enables new ones, such as Ultra-Reliable Low-Latency Communication (URLLC), which is essential for various real-time applications in today's fast-paced world. From autonomous vehicles to remote surgery, URLLC ensures minimal delay and utmost reliability, enhancing safety and efficiency. The requirements for URLLC in 5G networks are stringent, demanding ultra-low latency, high reliability, and scalability to support diverse applications.

In parallel to the 5G networks, Multi-access Edge Computing (MEC) is rapidly gaining traction and undergoing standardization to meet the demands of applications such as URLLC. MEC consists of the computing platform located in the proximity of the user at the edge of the network. MEC has the most prominent benefit of reducing latency and providing computation resources rather than relying solely on centralized data centers.

Vehicle-to-Everything (V2X) communication, an integral component of smart transportation systems, heavily relies on URLLC. 5G and MEC ensure seamless and real-time connectivity between vehicles, infrastructure, pedestrians, and other entities on the road. The development of a 5G-MEC testbed for testing V2X applications is essential to evaluate the performance and interoperability of edge computing solutions in dynamic vehicular environments and to explore the potential impact. The thesis contributions can be divided into three parts.

First, the thesis focuses on understanding the technical details of existing frameworks and specifications provided by standardized bodies for building a 5G-MEC testbed for V2X applications. This involves closely examining protocols, interfaces, and architectural guidelines required by the standardization entities. Additionally, the thesis looks into the individual hardware,

software, networking equipment, and communication interfaces needed for the testbed. By analyzing these technical aspects, the research activity aims to overview all the components to create a 5G-MEC testbed for V2X applications and how to tailor the available solutions (which can be open source) for these components to meet specific requirements of the testbed.

Secondly, in a dynamic scenario such as V2X, the orchestration of 5G-MEC applications is critical. A 5G-MEC system typically employs a distributed architecture where the network and computing resources are deployed in the proximity of the users. In such distributed environments, orchestration becomes essential to efficiently manage and coordinate the 5G-MEC applications across multiple edge nodes. This thesis aims to develop a testbed that includes the orchestration of MEC applications, which enables the effective allocation of data and network resources. The use of frameworks for the deployment, scaling, and life cycle management of MEC applications, such as Kubernetes (K8s), is explored. Furthermore, the thesis addresses the integration of K8s with the MEC framework, providing strategies for deploying orchestration in user mobility scenarios inherent to V2X communication.

Third, the research activity delves into the complexity of the joint allocation of data and network resources in 5G-MEC systems, evaluating strategies that optimize multiple performance objectives, such as latency, throughput, and Quality of Experience (QoE). The multi-objective optimization is particularly important in the provision of services with heterogeneous requirements as 5G does. The evaluation of the strategies is performed in a 5G-MEC testbed for V2X application.

Acknowledgements

I would like to express my heartfelt gratitude to all those who have contributed to completing this thesis.

First and foremost, my journey through this PhD wouldn't have been possible without my supervisors' unwavering support and guidance, Assoc. Prof. Gianfranco Nencioni and Assoc. Prof. Rosario Garroppo. I am deeply thankful for their insightful comments and suggestions, which significantly enhanced the quality of my work. Thank you for helping me, teaching me new things, sharing useful tricks, and going above and beyond. *Grazie mille*.

I'm grateful to Tom Ryen for his support and encouragement in teaching the course, an experience I'll always treasure. I thank my project colleagues and friends Annisa, Thilina, Muhidul, and Ruxandra for their feedback and suggestions throughout the years. Additionally, I extend my thanks to Jayachander and Theodor for their assistance.

I extend my gratitude to everyone I encountered at the University of Pisa, Italy, during my time there. I am particularly thankful to Assoc. Prof. Rosario Garroppo for his assistance throughout my stay; our discussions have helped me tremendously, including his efforts in resolving various challenges.

I wish to extend my heartfelt thanks to all my colleagues, including PhD, Master, and Bachelor students, and every single person with whom I crossed paths and had the privilege of meeting over the years. I have gained valuable insights from each interaction.

To all my friends, I am deeply grateful for the opportunity to share this journey with you. Thank you for the countless coffee breaks, weekend getaways, and hikes that have enriched our experiences together. Special shoutout to Qing, Arian, Ahmad, Herbert, Christin, Neel, Jiahui, Saul, Jorge, Nolwenn, Aitor, Hanish, Felix, Jacob, Boyu, and Jungwon.

Finally, I extend my heartfelt gratitude to my parents for their unwavering support. I appreciate their encouragement and genuine interest in my research, although occasionally, it might have sounded abstract and complex. And their continuous belief in my capabilities. I am grateful to my sister, Shreya, for her support and advice. I also want to thank all my friends back home and abroad for their support throughout this journey.

Thank you all wholeheartedly; I am sincerely grateful to all of you.

Prachi Vinod Wadatkar, August 2024

List of Abbreviations

3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
5GAA	5G Automotive Association
5G-DRIVE	5G HarmoniseD Research and TrIals for serVice Evolution
5G-VINNI	5G Verticals Innovation Infrastructure
AF	Application Function
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
AN	Access Network
APIs	Application Programming Interfaces
APS	Application Slice
AR	Augmented Reality
AS	Application Service
AUSF	Authentication Server Function
AWS	Amazon Web Services
CAD	Cooperative Automated Driving
CACC	Cooperative Adaptive Cruise Control
CCAM	Connected, Cooperative, and Automated Mobility

CMD	Command or Command Line
CN	Core Network
CPU	Central Processing Unit
CRI	Container Runtime Interface
C-V2X	Cellular Vehicle-to-Everything
DRL	Deep Reinforcement Learning
DNS	Domain Name System
E2E	End-to-End
ECC	Edge Cloud Computing
eMBB	Enhanced Mobile Broadband
eNB	Evolved NodeB
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
EU	Europe
EUD	Envelope Updated Design
FAST	Flexible and Low-Latency State Transfer
FOKUS	Fraunhofer Institute for Open Communication Systems
GPU	Graphics Processing Unit
gNB	Next Generation NodeB
HD	High Definition
IIoT	Industrial Internet of Things
ILP	Integer Linear Programming
IoT	Internet of Things
IaaS	Infrastructure-as-a-Service
IaC	Infrastructure as Code
IMT	International Mobile Telecommunication

ISG	Industry Specification Groups
ISP	Internet Service Provider
IT	Information Technology
ITS-G5	Intelligent Transportation Systems - G5
ITU	International Telecommunication Union
ITU-R	International Telecommunication Unit Radiocommunication
K8s	Kubernetes
KPIs	Key Performance Indicators
LCM	Life Cycle Management
LF Edge	Linux Foundation Edge
LL-MEC	Low Latency Multi-Access Edge Computing
LPWAN	Low Power Wide Area Network
LTE	Long-Term Evolution
LTE-M	Long-Term Evaluation of Machines
MANO	Management and Orchestration
MCDM	Multiple Criteria Decision Making
MCCe	MEC-caching-coexist
MDA	Multi-objective Dijkstra Algorithm
MDO	Multi-Domain Orchestrator
MEAO	Multi-access Edge Application Orchestrator
MEC	Multi-access Edge Computing
MEEP	Mobile Edge Emulation Platform
MEH	MEC Host
MEO	Multi-access Edge Orchestrator
MEPM	Multi-access Edge Platform Manager
MEP	MEC Platform

ML	Machine Learning
MODRL/HA	Multi-Objective Deep Reinforcement Learning/Heuristic Algorithm
MOAPSO-DP	Multi-Objective Hybrid Accelerated Particle Swarm Optimization and Dynamic Program
MOEAD_MEC	Multi-Objective Evolutionary Algorithm Based on Decomposition for MEC
MNOs	Mobile Network Operators
mMTC	Massive Machine-Type Communication
MOS	Mean Opinion Score
MOSP	Multi-Objective Shortest Path
MRAM	Multi-Objective Resource Allocation Method
MVNOs	Mobile Virtual Network Operators
NA	North America
NB-IoT	Narrowband Internet of Things
NRF	Network Repository Function
NGMN	Next Generation Mobile Networks
NR	New Radio
NS	Network Service
NSD	Network Service Descriptor
NSSF	Network Slice Selection Function
NF	Network Function
NFV	Network Functions Virtualization
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NEF	Network Exposure Function
NUC	Next Unit of Computing

OAI	Open-Air Interface
OBU	On-Board Unit
OMA	Open Mobile Alliance
OMNeT	Objective Modular Network Testbed
OSS	Operations Support System
OS	Operating System
OSM	Open Source MANO
OpenNESS	Open Network Edge Services Software
PAES	Pareto Archived Evolution Strategy
PaaS	Platform-as-a-Service
PCF	Policy Control Function
PND	Parameterized Network Design
PoA	Points of Access
PoPs	Points of Presence
QoS	Quality of Service
QoE	Quality of Experience
RAM	Random Access Memory
RAN	Radio Access Network
RAT	Radio Access Technology
REST	Representational State Transfer
RNI	Radio Network Information
RRV	Rapid Response Vehicle
ROs	Research Objectives
RQs	Research Questions
RSUs	Roadside Units
SBA	Service-Based Architecture

SCK	Service Construction Kit
SCP	Service Communication Proxy
SDK	Software Development Kit
SDN	Software-Defined Networking
SDR	Software-Defined Radio
SFCs	Service Function Chains
SLAs	Service Level Agreements
SMF	Session Management Function
TCP/IP	Transmission Control Protocol/Internet Protocol
T&L	Transport & Logistics
TSN	Time-Sensitive Networking
UDM	Unified Data Management
UDR	User Data Repository
UE	User Equipment
UPF	User Plane Function
URLLC	Ultra-Reliable Low-Latency Communication
VI	Virtualization Infrastructure
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2P	Vehicle-to-Pedestrian
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VIS	V2X Information Service
VIM	Virtualization Infrastructure Manager
VLC	VideoLAN Client
VM	Virtual Machine

VNF	Virtual Network Function
VNFD	Virtual Network Function Descriptor
VNFM	Virtual Network Function Manager
VNFV	Virtual Network Function Virtualization
VNFI	Virtual Network Function Infrastructure
VR	Virtual Reality
VRU	Vulnerable Road User
VUE	Vehicle User Equipment
YOLO	You Only Look Once

List of Publications

The main part of this dissertation is made up of the following published scientific papers:

- **Paper 1**

  5G-MEC Testbeds for V2X Applications
  Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni
  Published by the MDPI Journal of Future Internet, 2023

- **Paper 2**

  Experimental Evaluation of Handover Strategies in 5G-MEC Scenario by using AdvantEDGE
  Rosario G. Garroppo, Marco Volpi, Gianfranco Nencioni, Prachi V. Wadatkar
  Published in the Proceedings of the Mediterranean Conference on Communications and Networking (MeditCom), 2022

- **Paper 3**

  MEC Application Migration by using AdvantEDGE
  Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni
  Published in the Proceedings of the 17^{th} EAI International Conference on Tools for Design, Implementation and Verification of Emerging Information Technologies (TRIDENTCOM), 2022

- **Paper 4**

  Joint multi-objective MEH selection and traffic path computation in 5G-MEC systems
  Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni, Marco Volpi
  Published by the Elsevier Journal of Computer Networks, 2024

- **Paper 5**

MigraMEC: Hybrid Testbed for MEC App Migration

Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni

Published in the Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (MobiCom), 2023

Contents

Part I

Research Overview

Chapter 1

Introduction

Mobile data traffic is surging at a rapid pace, primarily driven by the popularity of video streaming services and the increment of video resolution. With the proliferation of multiple devices per user, the number of connections is steadily increasing. In parallel, the Internet of Things (IoT) needs networks capable of accommodating billions more devices. Even though the network operators need to meet these new requirements, they face increasing pressure to reduce operational costs, particularly for mobile network users. These challenges have to be addressed in order to allow the next generations of mobile communication technology to unlock new use cases and various industry applications [1].

The Fifth Generation (5G) of mobile network technology builds upon the advancements of the previous generations. Representing a significant leap forward, 5G boasts speeds that are 100 times faster than the Fourth Generation (4G) [2]. Beyond speed, 5G is envisioned as the backbone of a new era of connectivity, aiming to interconnect virtually everything, from machines and objects to various devices. 5G surpasses the speed and reliability of 4G networks, promising transformative impacts on internet usage, including access to applications. 5G aims to address anticipated surges in traffic and device connectivity by providing significantly higher data speeds, enhanced reliability, and reduced latency [3].

Figure 1.1 illustrates the diverse range of scenarios where 5G can be utilized, along with the corresponding network capabilities as outlined by the International Telecommunication Unit Radiocommunication sector (ITU-R) [4]. The potential of 5G is vast, with the capacity to support approximately 1 million devices within a one-square-kilometer area, facilitate mobility at

speeds of up to 500 kilometers per hour, and ensure highly dependable communication. 5G supports three main usage scenarios: enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low Latency Communication (URLLC), and massive Machine-Type Communications (mMTC)[5]. This versatility translates into a multitude of benefits across various sectors. For instance, eMBB enables new immersive experiences like Virtual Reality (VR) and Augmented Reality (AR), offering faster, more uniform data rates, lower latency, and reduced cost-per-bit. Moreover, 5G capabilities support mission-critical services, such as remote control of critical infrastructure and medical procedures, through ultra-reliable, available, low-latency links. In the automotive industry, the integration of 5G with machine-learning-driven algorithms promises enhanced traffic management, accident prevention, and real-time information sharing among vehicles and infrastructure entities like traffic lights. In order to meet the demands of various applications effectively, the integration of edge computing alongside 5G is essential [6]. For instance, while 5G offers latency as low as 1 millisecond, the incorporation of edge computing can efficiently and reliably facilitate the achievement of such low latency requirements [7].

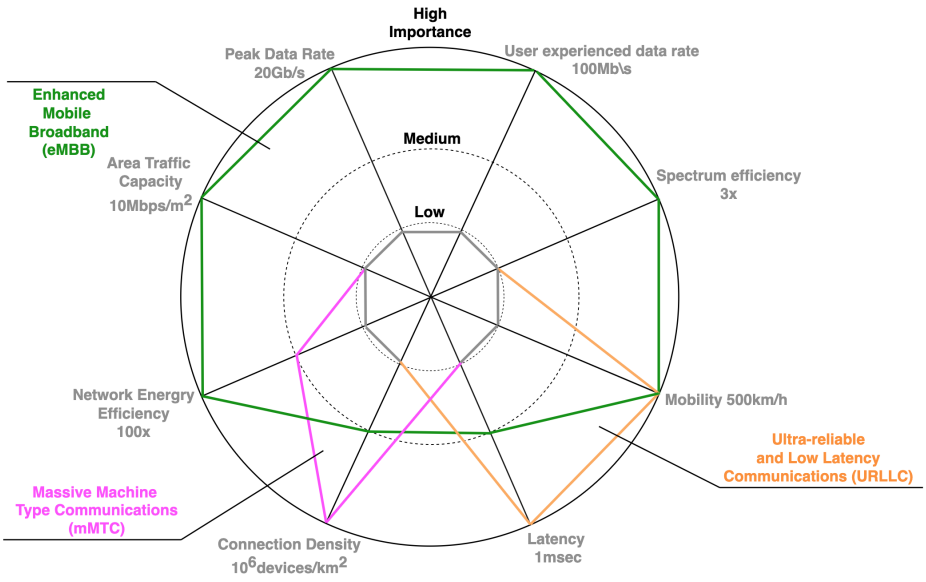


Figure 1.1: 5G usage scenarios [1].

Edge computing will be adopted by major enterprises by 2025 [8, 9], with

over 50% of enterprise-managed data expected to be created and processed outside traditional data centers or cloud environments. This transformation will be fueled by the rapid growth of IoT devices, which are projected to triple by 2030, leading to a dramatic increase in data generation. Furthermore, by 2027, 20% of large enterprises are anticipated to deploy edge management and orchestration solutions, marking a substantial rise from less than 1% in 2023 [10]. The advent of 5G technology coupled with edge computing has ushered in a new era of connectivity, promising ultra-low latency, high bandwidth, data security, and ubiquitous connectivity [11]. Edge computing revolutionizes network architecture by providing cloud computing capabilities and an IT service environment at the network edge [12]. By moving application hosts closer to end users, edge computing aims to reduce latency, enhance network efficiency, and elevate the overall customer experience [13]. This approach enables ultra-low latency, high bandwidth, and real-time access to data and radio network information, fostering rapid processing tasks and superior application performance. European Telecommunications Standards Institute (ETSI) [14] is one of the standardization bodies working towards interoperability and compatibility across diverse implementations by providing technical and architectural standards for the development of edge computing. ETSI refers to edge computing as Multi-access Edge Computing (MEC). ETSI's contributions drive innovation and foster a robust 5G-MEC ecosystem for various applications.

5G coupled with MEC technology may provide support for Vehicle-to-Everything (V2X) applications, which include various scenarios such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Network (V2N) communication, all of which rely on low-latency, high-bandwidth, and reliable connectivity [15]. V2X environments involve dynamic and complex network topologies, and the V2X ecosystem comprises a diverse array of stakeholders, including vehicle manufacturers, infrastructure providers, communication service providers, and application developers. Each of these stakeholders may have different requirements, objectives, and constraints when it comes to V2X testing [16]. Given the diverse stakeholders and their unique requirements in V2X testing scenarios, developing a dedicated 5G-MEC testbed for V2X applications is essential.

1.1 Problem Statement

Despite the individual advancements in 5G and MEC technologies, the creation of a testbed for V2X applications integrating both 5G and MEC poses significant challenges. The challenges include selecting and integrating the various hardware and software of the 5G-MEC testbed to provide the needed functionalities for V2X scenarios. There are existing open-source solutions for implementing 5G-MEC functionalities, which require integration and customization.

In addition to building a 5G-MEC testbed, a crucial aspect is the design and implementation of an orchestration framework for MEC applications and services deployment within the 5G-MEC ecosystem. The orchestration of the deployment and operations of MEC applications and services is essential to ensure efficient resource management. Dynamic scaling capabilities enable the testbed to adapt in real-time to fluctuations in demand, allowing for the seamless expansion or contraction of MEC resources to match the varying needs of V2X applications, including the impact of user mobility.

In a 5G-MEC system, the allocation of data and network resources presents a significant challenge. This entails balancing multiple objectives like minimizing latency, enhancing energy efficiency, and maintaining service quality. The evaluation of various allocation strategies and their performance in a 5G-MEC testbed across different network conditions and traffic loads is essential to understand their real potential. Additionally, the user mobility typical of a V2X environment introduces complexity since the MEC applications would need to be migrated from one MEC host to another in order to maintain a low latency. An adaptive resource allocation algorithm is necessary to dynamically optimize the resource allocation based on user mobility patterns, network conditions, and service requirements, ensuring a seamless user experience of V2X applications.

The development of a dedicated 5G-MEC testbed for V2X applications creates the following challenges: (i) Integration of 5G networks and MEC infrastructure by using open-source tools offers flexibility and cost-effectiveness, but open-source solutions may lack comprehensive support for 5G-MEC integration, resulting in compatibility issues, limited feature sets, and suboptimal performance. (ii) Efficient orchestration and management of MEC applications are critical for optimizing application performance. (iii) Effective allocation of data and network resources with multiple objectives is essential for ensuring low-latency, high-throughput communication in 5G-MEC environments.

1.2 Project Contributions

The primary focus of this thesis is the orchestration of MEC services and applications within a 5G-MEC testbed, considering user mobility in the context of V2X applications. Given the complexity of developing 5G-MEC extensive system, which requires considerable resources and time, this thesis strategically narrows its focus to establishing a hybrid testbed as a foundational step. This testbed serves as an initial platform for exploring orchestration in 5G and MEC environments, offering a practical approach to studying these complex systems.

The research specifically explores the orchestration of MEC services, in order to enable the efficient resource allocation, service continuity, and low latency—factors that are essential for meeting the real-time demands of V2X communications. In addition to orchestration, the thesis addresses user mobility support, a critical aspect for V2X applications, where the high-speed movement of vehicles necessitates seamless handovers and service migration across MEC nodes. The 5G core and other network components are simulated within this controlled environment, allowing for comprehensive testing and validation of the proposed orchestration strategies without the need for a full-scale 5G deployment. This approach provides a solid foundation for future enhancements and real-world implementations.

The main contributions of the thesis are concluded in five published papers. The papers are listed and briefly summarized as follows:

I. **Paper 1:**

   Prachi V. Wadatkar, Rosario G. Garroppo and Gianfranco Nencioni, **"5G-MEC Testbeds for V2X Applications"**, MDPI Journal of Future Internet, 2023.

   This paper presents insights into the framework of 5G-MEC systems and offers a comprehensive overview of testbed implementations tailored for V2X applications. It evaluates existing testbeds within this domain, categorizing them according to their use cases and assessing their usage complexity based on factors such as replication and open-source accessibility. Additionally, the paper outlines open-source solutions for the software-defined functions of 5G and MEC, along with simulative tools for testing real-world V2X scenarios. Finally, the study concludes by identifying research gaps crucial for the advancement of testbed development.

II. **Paper 2:**

Rosario G. Garroppo, Marco Volpi, Gianfranco Nencioni, Prachi V. Wadatkar, **"Experimental Evaluation of Handover Strategies in 5G-MEC Scenario by using AdvantEDGE"**, Published in the Proceedings of the Mediterranean Conference on Communications and Networking (MeditCom), 2022.

Building upon the open-source solutions explored in Paper 1, Paper 2 focuses on a particular scenario utilizing the AdvantEDGE tool, which is a Mobile Edge Emulation Platform (MEEP). The setup involves a specialized testbed incorporating 5G-MEC and the video streaming application. This configuration enables the examination of performance metrics related to multiple Radio Access Technology (multi-RAT) technologies different handover strategies developed through a multi-objective approach.

III. **Paper 3:**

Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni, **"MEC Application Migration by using AdvantEDGE"**, Published in the Proceedings of the 17^{th} EAI International Conference on Tools for Design, Implementation and Verification of Emerging Information Technologies (TRIDENTCOM), 2022.

In Paper 2, the focus is on achieving optimal Quality of Experience (QoE) for end-users by orchestrating handovers between multi-RATs to access MEC applications hosted on a single MEC Host (MEH). Paper 3 explores further enhancing QoE by introducing multiple MEHs into the scenario. Additionally, MEH handovers for seamless migration of MEC applications are addressed using the ETSI-MEC location service Application Programming Interface (API) and integration of a prototype solution based on Docker and Kubernetes (K8s) in the specifically designed 5G-MEC testbed environment.

IV. **Paper 4:**

Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni, Marco Volpi, **"Joint multi-objective MEH selection and traffic path computation in 5G-MEC systems"**, Published by the Elsevier Journal of Computer Networks, 2024.

In Paper 4, the focus is on joint MEH selection and traffic path computation. The paper proposes a procedure to define a graph considering network and application performances in order to use the Multi-objective Dijkstra Algorithm (MDA) to solve the problem. MDA computes the Pareto front of the Multi-Objective Shortest Path (MOSP) model by using the defined graph. A hybrid testbed evaluates MDA performance, including simulative, emulative, and experimental parts. Additionally, a controller integrates MDA with the 5G-MEC system in the testbed, handling input retrieval and MDA output application across network and application layers.

V. **Paper 5:**

Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni, **"MigraMEC: Hybrid Testbed for MEC App Migration"**, Published in the Proceedings of the 29$^{\text{th}}$ Annual International Conference on Mobile Computing and Networking (MobiCom), 2023.

In Paper 5, a hybrid testbed named as "MigraMEC" addresses the challenge of MEC application migration in the presence of user mobility. The paper presents a hybrid testbed architecture that combines network simulation, user mobility emulation with AdvantEDGE, and an extended K8s for physical MEHs deployment. The demonstration underscores the significance of multiple MEHs and seamless migration for sustaining high user experience, particularly illustrated through a video streaming service.

1.3 List of Other Works

VI. **Paper 6:**

Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni, **"Evaluating the Performance of a Real-Time VRU Detection System Using Edge Devices"**, To be Submitted.

In Paper 6, the focus is on real-time Vulnerable Road User (VRU) detection MEC application. The paper proposes a hybrid system where end-users provide streams of traffic video to the nearest MEH, emulating a real-world scenario. AdvantEDGE is utilized to emulate

the network environment. MEH processes the video by incorporating the well-known You Only Look Once (YOLO) model for object detection. As MEH detects VRU, the end-user receives a warning.

1.4 Outline of the Thesis Structure

The thesis comprises two separate parts. Part I serves as an introductory framework, offering an overview of the project's outcomes and delving into the core topic of the thesis. Part II encompasses the compilation of included papers. Within Part I, the following chapters are included:

- **Chapter 2** provides background information relevant to the domain of the thesis.
- **Chapter 3** discusses related works within the area of the thesis's research contributions.
- **Chapter 4** delves deeply into the research contributions by outlining research questions and objectives. The chapter also establishes links between these questions and the included papers, along with descriptions of the research methodologies used in each paper.
- **Chapter 5** concludes by giving an overview of the main findings and discussing potential future directions.

Chapter 2

Background

This chapter overviews 5G and MEC technologies and their requirements. Additionally, it summarizes the utilization of open-source solutions for implementing software-defined 5G-MEC functionalities and the management and orchestration of MEC applications. Finally, the chapter presents diverse V2X use cases and discusses how 5G-MEC can contribute to supporting these V2X applications.

2.1 5G Overview

The 5G infrastructure encompasses the physical components constituting a 5G network, enabling its advanced functionalities. The 5G infrastructure exploits network slicing, which allows to divide the physical network into multiple virtual networks to offer customized services and connectivity for diverse users and applications. The 5G Core Network (CN) is characterized by its virtualized and cloud-native architecture, departing from the hardware-centric approach of previous generations, thereby affording enhanced flexibility, scalability, and programmability. Overall, the 5G infrastructure adopts a more distributed, software-defined, and adaptable framework than preceding mobile networks, empowering it to accommodate a broad spectrum of novel applications and use cases (eMBB, URLLC, and mMTC).

2.1.1 Key Characteristics and Capabilities

5G represents a significant leap forward in mobile telecommunications and is characterized by faster speeds, lower latency, greater capacity, and enhanced connectivity compared to 4G. The key characteristics and capabilities of 5G are the following ones [5]:

- *High Speeds* – 5G networks offer peak data rates up to 20 Gbps, enabling ultra-fast downloads, high-definition video streaming, and real-time gaming.
- *Low Latency* – 5G ensures real-time responsiveness crucial for applications like VR, AR, autonomous vehicles, and remote surgery by reducing the latency to 1 millisecond, which is imperceptible to humans.
- *High Capacity* – 5G supports significantly more connected devices, which is essential for IoT proliferation and smart city infrastructure integration, fostering innovation and sustainability.
- *Enhanced Connectivity* – 5G ensures improved signal strength and coverage, enhancing connectivity reliability in urban and indoor environments by utilizing advanced antenna technologies.
- *Network Slicing* – 5G allows operators to create virtual networks optimized for specific use cases, ensuring efficient data transmission and resource allocation for diverse applications by introducing network slicing.

Comparison of essential features between IMT-Advanced (4G) and IMT-2020 (5G), where IMT stands for International Mobile Telecommunication, as per ITU-R M.2083 criteria presented in Figure 2.1

The evolution of telecommunication architecture now revolves around 5G technology, which introduces modular Network Functions (NFs) for both the control plane and user plane across the Access Network (AN) and CN. These NFs offer specialized network functionalities to suit diverse application requirements. 5G aims to serve various vertical sectors and industries, encompassing machine-to-machine and human-to-human communication. Moreover, 5G integrates MEC to reduce latency and enhance network efficiency. ITU-R, in Recommendation ITU-R M.2083, outlines the main usage scenarios for IMT beyond 2020 as follows [1, 5]:

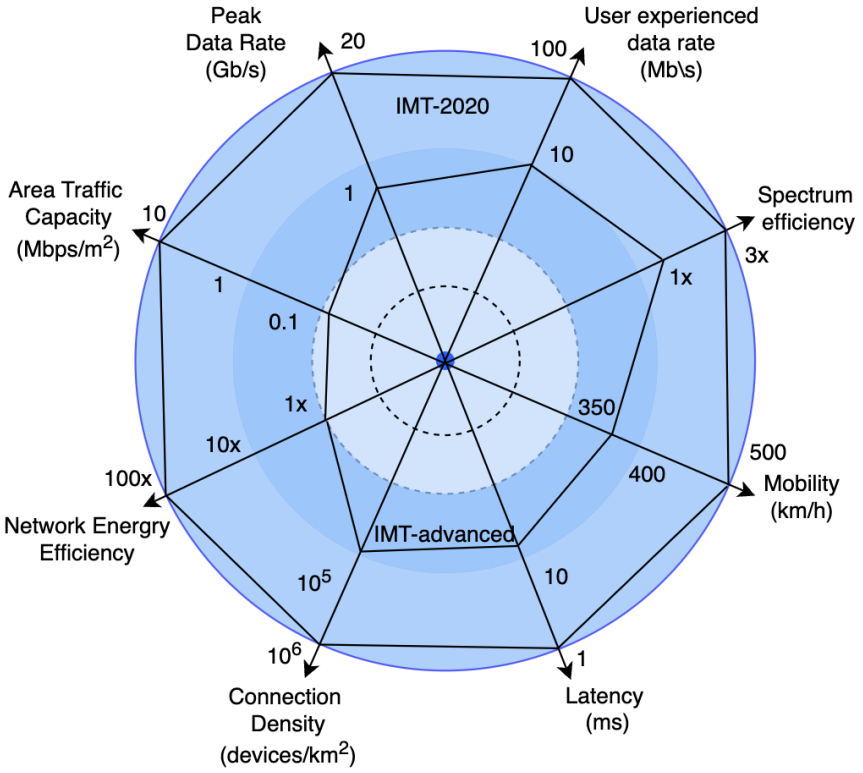


Figure 2.1: Features between IMT-Advanced and IMT-2020 [1].

- *eMBB* – Meeting demands for increased data rates, high user density, and extensive traffic capacity in hotspots and seamless coverage scenarios while enhancing user data rates for high mobility situations. Additionally, eMBB offers higher data rates than 4G and moderate latency improvements to support applications like AR/VR media and UltraHD streaming.

- *mMTC* – Tailored for IoT, requiring low power consumption and data rates to accommodate many interconnected devices. mMTC also provides connectivity for many devices, replacing Low Power Wide Area Network (LPWAN) technologies such as Narrowband Internet of Things (NB-IoT) and Long-Term Evolution for Machines (LTE-M).

- *URLLC* – Designed to support safety-critical and mission-critical

applications, necessitating specific key capabilities specified in ITU-R M.2083. URLLC further requires deploying the 5G core to achieve End-to-End (E2E) latency reduction, supporting applications like remote control and autonomous driving with stringent reliability requirements.

2.1.2 Comprehensive Overview of 5G Architecture

The 5G system architecture marks a paradigm shift towards a service-based approach, delineating the interaction between NFs in distinct manners. With the deployment of 5G systems, the utilization of softwarization technologies has become increasingly prevalent. These technologies are poised to revolutionize the capabilities of 5G, enabling it to:

- Facilitate independent scalability, deployment, and segregation of User Plane Functions (UPFs) from control plane functions.
- Enhance flexibility in deployment and facilitate modifications to function designs, optimizing network slicing efficiency. Network slicing, in turn, offers an E2E virtual network encompassing network, compute, and storage functions. Network slicing enables the customization of network resources to meet the diverse requirements of different applications, users, or services within a single physical network infrastructure.
- Foster direct interaction between NFs, minimizing dependencies between CN and AN components.

Figure 2.2 depicts the non-roaming 5G system reference architecture. The following NFs [3] are shown in Figure 2.2:

- *Access and Mobility Management Function (AMF)* handles user authentication, mobility management, and resource allocation within the 5G network.
- *Session Management Function (SMF)* establishes and manages data sessions, assigns IP addresses, and ensures Quality of Service (QoS) for user sessions.
- *UPF* performs packet forwarding, filtering, and policy enforcement for user data traffic.

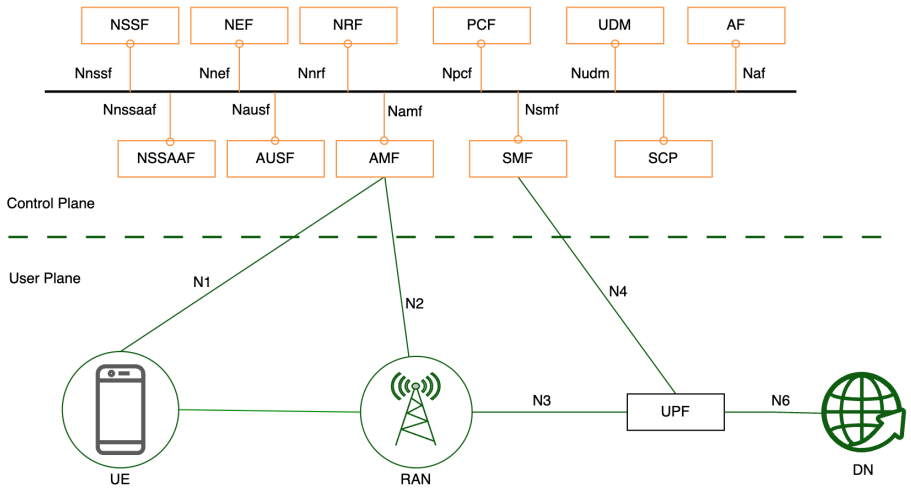


Figure 2.2: 5G system architecture [3].

- *Network Repository Function (NRF)* centralizes all 5G NFs and provides a standardized API for registration and discovery, crucial for implementing the new Service-Based Architecture (SBA) in the 5G core.
- *Policy Control Function (PCF)* simplifies policy development and implementation in the 5G network, designed with cloud-native principles to monetize and optimize 5G services.
- *Service Communication Proxy (SCP)* enhances CN operations by offering routing control, resiliency, and observability, leveraging IT service mesh to address challenges posed by the new SBA in the 5G core.
- *Network Slicing Selection Function (NSSF)* selects the most suitable network slice for requested services in the diverse 5G environment with multiple service offerings.
- *Unified Data Management (UDM) and User Data Repository (UDR)* cloud-native UDM, similar to LTE Home Subscriber Server (HSS), handles authentication credentials creation, access authorization, and roaming, supported by UDR for user data retrieval.
- *Authentication Server Function (AUSF)* executes 5G authentication

and Key Agreement method 5G AKA, managing hidden or privacy-protected subscription identifiers during registration.

- *Network Exposure Function (NEF)* is a component that facilitates secure and controlled access to network data and services, enabling efficient communication between external applications and the 5G network.

2.2 Multi-access Edge Computing (MEC)

MEC emerges as a fundamental component of 5G ecosystems, positioning computing resources and capabilities at the network edge [12], significantly reducing latency and enhancing bandwidth for diverse end-users such as vehicles, pedestrians, and IoT devices. By deploying within the Radio Access Network (RAN) or transport network, MEC facilitates efficient offloading of computing tasks from User Equipment (UE), eliminating the need for resource-intensive functions on the user side [3]. MEC standardization via ETSI's Industry Specification Groups (ISG) enables the creation of agile virtualized environments for edge services across verticals. This integration becomes essential amid ongoing demand for computational and URLLC applications within 5G networks, addressing concerns regarding end-user limitations and storage constraints. MEC deployment within the RAN ensures real-time processing capabilities, mitigating challenges posed by unprecedented traffic volume. While fog computing and cloudlet offer alternative architectures, MEC stands out for deploying cloud computing capabilities within the RAN, enhancing end-user experiences and enabling seamless integration of computational-intensive applications within 5G environments.

2.2.1 Understanding the MEC Framework

The MEC architecture outlines how the MEC system allows MEC applications to run smoothly and efficiently across different types of networks. ETSI defines the MEC reference architecture that includes one or more MEC Hosts (MEHs) and the MEC management entity as shown in Figure 2.3. Next, the different parts of the architecture are described [17].

MEH comprises the MEC Platform (MEP) and a Virtualization Infrastructure (VI) encompassing computing, storage resources, and networking

dedicated to MEC applications. Within the VI, the data plane executes traffic regulations received by the MEP. MEH manages the deployment of MEC applications, services, and ANs. Virtual Machines (VMs) maintain the isolation between the operations conducted and deliver network service.

MEP is a component designed to execute the necessary functionalities for running MEC applications within a specific VI. It configures Domain Name System (DNS) proxy and server settings and furnishes an environment for providing and utilizing MEC services. Moreover, the MEP administrates access to storage resources and time-of-day information.

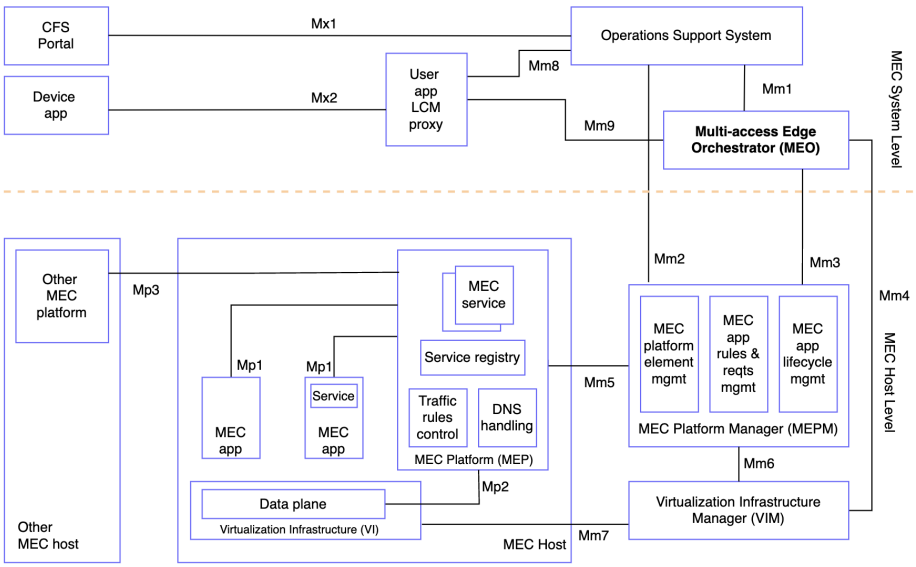


Figure 2.3: MEC reference architecture [17].

MEC Management: MEC management comprises two key components: MEC system-level management and MEC host-level management. MEC system-level management interacts with both MEC host-level management and the AN. MEC system-level management includes the following elements:

- *Multi-access Edge Orchestrator (MEO)* is the core element within the MEC system, where the information on MEHs, MEC services, computing resources, and network topology is known. MEO determines the appropriate MEH selection for users and executes system-level tasks with support from the Operating Support System (OSS). Additionally,

it facilitates application termination, relocation, and instantiation as necessary.

- *OSS* receives requests via the Customer Facing Service (CFS) portal for application instantiation or termination and provides decisions on these requests. MEO handles these requests for further processing. When applicable, OSS also manages requests for application relocation between the MEC system and the cloud.
- *User Application Lifecycle Management Proxy* facilitates device applications in requesting onboarding, instantiation, termination, and relocation of user applications within the MEC system while also managing communication with OSS and MEO for further processing.

MEC host-level management includes the following elements:

- *MEC Platform Manager (MEPM)* is responsible for managing the life cycle of applications, including informing the MEO about relevant application-related events and providing element management functions to the MEP. It manages application rules and requirements, such as service authorizations, traffic rules, and DNS configuration, while also receiving fault reports.
- *Virtualization Infrastructure Manager (VIM)* component allocates, manages, and releases virtualized compute, storage, and networking resources within the VI. It prepares the infrastructure for running software images, supports rapid application provisioning, and collects/reports performance and fault information about virtualized resources.

The architecture illustrates reference points connecting various elements, with Mp representing MEP reference points, Mm representing MEC management reference points, and Mx representing external entities reference points.

2.2.2 Role of MEC in 5G Networks

The integrated deployment of the MEC system within a 5G network is depicted in Figure 2.4, wherein the MEO, acting as an AF, interacts with the NEF or directly with target 5G NFs. At the MEH level, MEP engages with

various 5G NFs within the data network, while NEF, typically centralized in CN, can also deploy at the edge for swift service access. ETSI MEC is deployed on the N6 logical reference point interface, i.e., in a data network external to the 5G system, enabled by flexibility in the UPF location. In addition to MEC apps, the distributed MEH can accommodate a message broker as an MEP service and another MEP service to steer traffic to local accelerators [18].

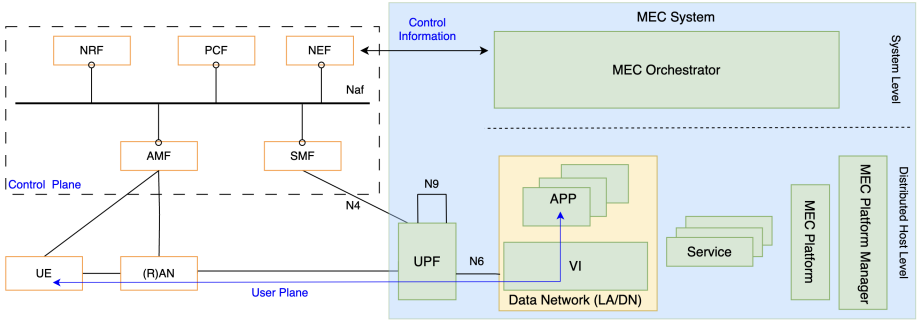


Figure 2.4: Deployment of MEC integrated within 5G network [18].

User mobility management, overseen by the AMF in a mobile communications system, is relevant to MEC. As users move between different network areas, the AMF ensures seamless handover and connectivity, which is essential for uninterrupted MEC services. For instance, when a user transitions between cellular towers or switches from one RAT to another, the AMF facilitates the handover process to maintain continuity in MEC applications and services [18]. In V2X applications, where low latency is critical, the integration of user mobility data can enhance the MEC application migration and MEH handover processes. Specifically, predictive models that analyze user movement patterns, including speed, direction, and trajectory, can be leveraged to anticipate when and where a handover or migration is necessary. This proactive approach allows for the pre-allocation of resources at the target MEH or access point, minimizing latency and reducing the risk of service degradation during the transition. Moreover, by utilizing relevant MEC APIs, the MEC system can gain a more granular understanding of user trajectories, enabling finer control over the placement of UPFs and other critical network functions relative to the user's predicted path.

Similarly, the SMF plays a crucial role in MEC deployments by managing

sessions, allocating and managing Internet Protocol (IP) addresses, and ensuring connectivity for users accessing MEC services. Efficient session management is essential for maintaining the integrity and performance of MEC applications, particularly in scenarios where users move between different network slices or access points within a MEC-enabled network [18]. Integrating user movement predictions into the SMF's decision-making process can lead to optimized session continuity, as the SMF can preemptively adjust session parameters or reroute traffic based on anticipated user location. This ensures that session management remains robust even in highly dynamic environments, thereby maintaining the QoS required by latency-sensitive applications like V2X [19].

Benefits of MEC in 5G Environments

MEC offers several advantages in 5G environments, revolutionizing network capabilities and user experiences [20]. Here are some key advantages of MEC in 5G:

- *Low Latency* – MEC minimizes latency in 5G by deploying computing resources closer to end-users at the edge. Real-time data processing enables ultra-low latency responses, which is critical for applications within the automotive industry.
- *Scalability* – MEC distributed architecture allows dynamic resource scaling based on demand, optimizing resource utilization in 5G networks. This ensures optimal network performance amidst increasing device connections and data-intensive applications.
- *Improved QoS* – By processing data locally, MEC reduces network congestion and enhances bandwidth utilization in 5G. This results in faster data transmission and superior user experiences, particularly in latency-sensitive applications like video streaming and online gaming.
- *Enhanced Security* – MEC strengthens security in 5G by implementing localized security measures at the network edge. This reduces the risk of data breaches and unauthorized access, ensuring secure communication between edge devices and applications.
- *Support for Edge Intelligence* – MEC enables real-time data analytics and decision-making at the edge, facilitating the deployment of edge

intelligence and Machine Learning (ML) algorithms in 5G networks. This empowers personalized services, predictive maintenance, and innovative applications across various industries.

2.2.3 MEC APIs Functionality

ETSI's ongoing efforts in standardizing MEC APIs aim to foster smooth integration with third-party solutions of MEC elements. By providing RESTful APIs for designated services, ETSI ensures the availability of crucial information and establishes a robust framework for adapting to dynamic environmental shifts. These standardized APIs are also vital in securing user data transfer and safeguarding sensitive information across diverse MEC environments. The RESTful design of MEC-specific APIs builds upon the widely accepted concept of RESTful programming, leveraging the HTTP protocol for interaction between remote entities. This stateless design, guided by Representational State Transfer (REST), aligns with industry best practices and enables efficient communication between MEC applications and the underlying infrastructure [21].

Figure 2.5 illustrates the function of MEC APIs and the interaction among software-defined MEC functionalities. MEC services, such as Radio Network Information (RNI), Location, Bandwidth, and UE Identity, are accessible to both the MEP and authorized MEC applications. The APIs can be accessed in YAML and JSON format that comply with the OpenAPI standards. Most relevant MEC APIs to support 5G-MEC testbeds are as follows [22]:

- *RNI API* [23] delivers real-time data from the RAN to MEC applications, empowering dynamic service optimization based on current radio conditions. Accessible via RESTful queries or a pub/sub mechanism, it supports many use cases outlined in ETSI Group Specification (GS) MEC 002 [24] and ETSI GS MEC 012.

- Location API [25] enables MEC applications to access precise location data, facilitating tailored services and dynamic content delivery based on geographical or logical coordinates. Utilizing RESTful APIs defined by the Open Mobile Alliance (OMA) and detailed in ETSI GS MEC 013, it supports geolocation queries, device tracking, and subscription-based updates for optimized service delivery and network analytics.

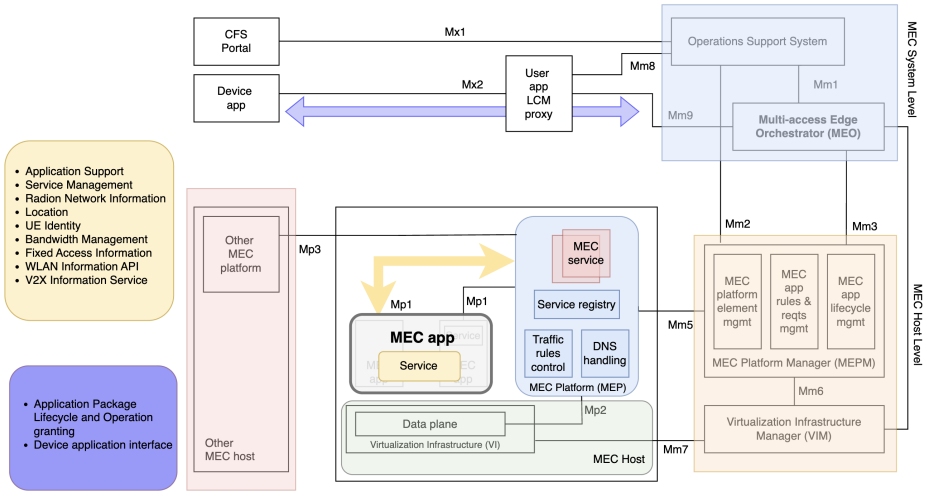


Figure 2.5: Role of MEC API [21].

- *UE Identity API* [26], outlined in ETSI GS MEC 014, enables MEC applications to manage the association of IP traffic flows with specific UEs using externally defined tags, enhancing user privacy and identity protection. By allowing registration and deregistration of tags representing UEs, the API facilitates efficient traffic filtering in the MEP, directing authorized traffic directly to designated destinations such as local enterprise networks, as described in ETSI GS MEC 011 [27].
- *V2X Information Service (VIS) API* [28] supports V2X interoperability by enabling the MEC VIS to gather V2X data via PC5 interface which is essential for direct V2X communication without relying on cellular networks, expose it to MEC apps, and facilitate secure communication with 3rd Generation Partnership Project (3GPP) CN functions, ensuring seamless operation across diverse MEC systems and potentially leveraging other MEC APIs for enhanced functionality.

MEC APIs play a crucial role in enhancing MEC services by facilitating precise and dynamic management of network resources and user experiences. For instance, the ETSI MEC location API provides critical location data that enables MEC applications to tailor services based on users geographical positions. However, comprehensive understanding and prediction

of user movements require additional data, such as speed, direction, and timestamps. VIS API is instrumental in this context, integrating V2X data to support accurate tracking and prediction of user trajectories. Leveraging such detailed movement data allows MEC systems to proactively plan MEH handovers, optimizing service delivery and reducing latency. Integrating these advanced APIs into MEC services results in a more responsive and efficient network, thereby enhancing overall performance and user satisfaction.

2.3 Softwarization Technologies

Software-based technologies are experiencing rapid expansion as the preferred network paradigms among service and network providers. This approach to software technologies encompasses designing, implementing, managing, and deploying services/network equipment through software programming [29]. Softwarization technologies aim to offer 5G services with cost-effectiveness and agility while also focusing on achieving E2E latency requirements. Furthermore, softwarization technologies aim to enhance the quality of user experience.

2.3.1 Software-Defined Networking (SDN)

SDN is an innovative networking paradigm that separates the control and data planes, enabling centralized network management and programmability [30]. SDN strives to enhance network management, programmability, and flexibility by simplifying network access by separating the control plane from the data plane [31]. This separation is facilitated by a logically centralized network intelligence known as the SDN controller, while the data plane comprises network devices responsible for packet forwarding. The SDN architecture's control plane encompasses the SDN controller and northbound and southbound interfaces. The northbound programming interface offers an abstract view of the network and its policies, whereas the southbound interface facilitates information exchange between the control plane and the data plane [32].

SDN addresses various challenges in traditional networks by offering dynamic configuration, enhanced security, and efficient resource management. Leading tech companies like Google and Facebook have already implemented SDN in their data centers. SDN is evolving to cater to diverse

network types, from data centers to IoT and Internet Service Provider (ISP) networks, providing solutions for unpredictable traffic patterns and rapid reconfiguration [33]. Overall, SDN's programmability and centralized management make it a promising architecture for modern networks, albeit requiring continuous advancements in security and efficiency.

SDN significantly enhances the flexibility and efficiency of MEC in 5G/Sixth Generation (6G) networks. Integrating SDN with MEC architectures, as proposed by [34], enables centralized global control management for efficient resource orchestration and service mobility. SDN simplifies network and service mobility management within resource-constrained MEC servers, addressing challenges in diverse service requests and dynamic environments like vehicular communications [35].

2.3.2 Network Function Virtualization (NFV)

NFV enables network operators to deliver network services using virtual infrastructure. NFV facilitates the transition from hardware-based proprietary NFs to software-based NFs deployed on off-the-shelf hardware [36]. This approach offers cost efficiency and leverages the timely benefits of cloud computing. NFV seeks to mitigate resource constraints and complex hardware integration by leveraging standard IT virtualization to consolidate NFs onto industry-standard servers, facilitating flexibility and scalability without additional hardware installations. NFV complements SDN, offering a synergistic approach to network optimization and management, although their implementation is not necessarily interdependent [37].

The NFV architecture [38] comprises the following components:

- *Virtual Network Functions (VNFs)* – These execute the software implementations, including NFs deployed in virtual environments.
- *NFV Infrastructure (NFVI)* – This component supports the execution of VNFs by providing virtual resources. NFVI encompasses infrastructure components such as compute, storage, network, and their respective hardware components.
- *Management and Orchestration (MANO)* – MANO offers the framework for NFV management, covering the LCM and orchestrating physical and software resources to deliver network services. MANO elements include NFV Orchestrator (NFVO) and VNF Manager (VNFM). NFVO is responsible for managing the lifecycle of VNFs.

It handles tasks such as VNF instantiation, scaling, monitoring, and termination. VNFM manages the VNF lifecycle, from instantiation to termination, communicating with NFVO for orchestration and with VIM for resource management, ensuring VNFs adhere to service requirements.

2.3.3 MEC Deployment in NFV Environment

In the future, Mobile Network Operators (MNOs) will utilize NFV to provide network services. MNOs are integrating VI to consolidate network elements, MEC applications, and MEC components atop this infrastructure. The data plane manages traffic routing among applications, various services, ANs, and the MEP within the VI. Configuration of the data plane is facilitated through the Mp2 reference [39]. The VI is a central component of both NFV and MEC architectures.

In the NFV architecture variant incorporating MEC, specific functional blocks replace existing components to cater to MEC-specific functionalities. The MEC Application Orchestrator (MEAO) and the MEPM are introduced, each with distinct responsibilities. The MEAO, similar to the previous MEO, authorizes the management of MEC applications but delegates this task, along with managing MEC application VNF packages, to the NFVO. The MEP manager - NFV (MEPM-V), resembling the MEPM, relinquishes direct action on LCM and resource monitoring to a VNFM, streamlining the process and ensuring efficiency. The NFVO, defined by ETSI standards, is entrusted with managing the lifecycle of network services, treating each MEC application instance as a VNF instance. Similarly, VNFM components, responsible for the MEP and application LCM, adhere to NFV standards, ensuring uniformity and compatibility across deployments. Flexibility is maintained, allowing for multiple instances of the VNFM (MEC application LCM) block and the possibility of it being the same as the VNFM (MEP LCM), enhancing scalability and adaptability in diverse network environments [40]. Deploying MEC within the NFV environment offers significant benefits, such as low latency and scalability. NFV ensures scalability, while MEC contributes to reducing latency. Figure 2.6 depicts the MEC reference architecture within an NFV environment.

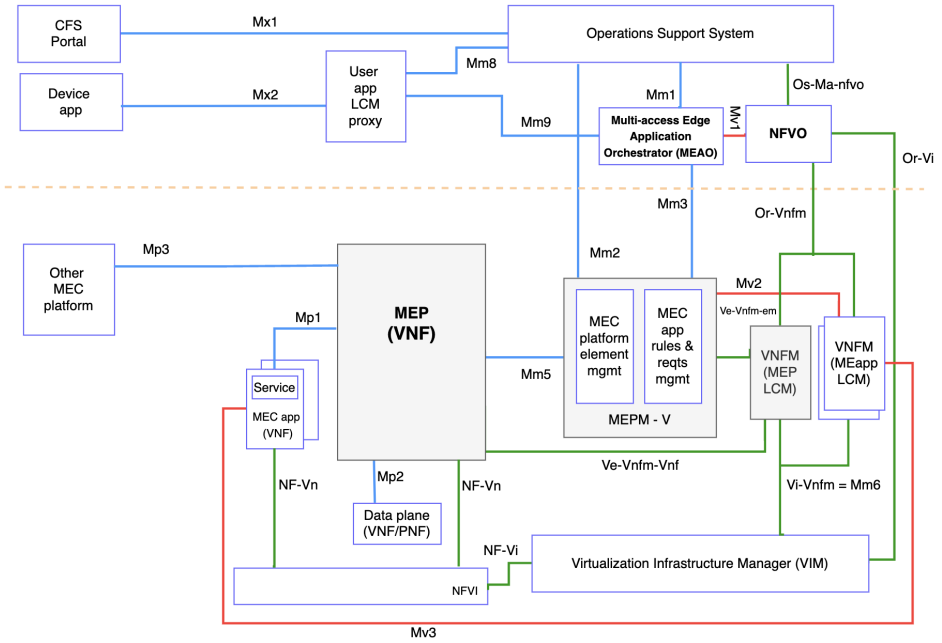


Figure 2.6: MEC reference architecture in NFV environment [40].

2.4 Introduction to Open Source Solutions

Various open-source solutions have emerged to facilitate the development of 5G-MEC software elements described in Section 2.2.1. The following table 2.1 presents a comprehensive compilation of these tools, organized according to their software-defined functionalities pertinent to 5G and MEC. Despite the availability of such solutions, a significant challenge persists in ensuring seamless integration and development of relevant APIs for conducting experimental studies.

2.4.1 MEC Ecosystem Solutions

In addition to the array of MEC solutions outlined in Table 2.1, the MEC ecosystem offers another invaluable resource: an ETSI repository for 3rd party solutions. The MEC ecosystem serves as a repository of relevant information regarding the endeavors of the ETSI ISG MEC, aiming to accelerate the progression of the MEC ecosystem. Presented below are concise descriptions of several state-of-the-art MEPs and technologies, each

Table 2.1: List of open-source solutions available in 5G-MEC.

Category	Tool/Framework	Key Features & KPIs
5G Infrastructure	OpenAirInterface (OAI) [41]	Support for 4G and 5G functionalities, Integration with UERANSIM [42] for signal transmission, Low-latency communication
	Open5GCore [43]	compliance with 3GPP standards, Lightweight deployment on Raspberry Pi or computing servers, Scalability and flexibility
	Open5GS [44]	Support for 4G and 5G CN deployments, Integration with UERANSIM for multiple UE support, flexibility and scalability
	Free5GC [45]	Basic 5G CN functionalities, Potential for future compliance with 3GPP standards, VM deployment
	Simu5G [46]	Simulation of 5G NR scenarios, Resource allocation, and management testing, Compatibility with ETSI MEC system model
	UERANSIM [42]	Simulation of 5G RAN protocols, testing of multiple UE scenarios, Integration with OAI for full 5G network simulation
MEC Solutions	OpenNESS [47]	Compliance with ETSI MEC architecture, support for high-performance data planes, integration with orchestration tools and VIMs
	EdgeX Foundry [48]	Docker-based containerization, Support for IoT edge computing, MQTT broker for device and data management
	LightEDGE [49]	Compliance with ETSI MEC architecture, Microservice architecture on K8s, Integration with Open5GS and srsRAN solutions
	AdvantEDGE [50]	Compliance with ETSI MEC architecture, Sandbox creation for scenario deployment, Support for V2X services and network traffic control
	LL-MEC [51]	Support for low-latency MEC applications, REST API for MEC services, Integration with SDN principles and OpenFlow protocol
Virtual Infrastructure and Manager	Docker [52]	Lightweight containerization, Compatibility with K8s, Accelerated application deployment and management
	K8s [53]	Container orchestration and scaling, Efficient management of containerized applications, Integration with SDN principles
	OpenStack [54]	management of virtualized computing, storage, and networking resources, Deployment flexibility, scalability, and resource allocation
Orchestration	OSM [55]	NFV orchestration and VNF management, Compatibility with VIMs like OpenStack, Comprehensive hardware resource requirements
	OpenBaton [56]	NFV orchestration framework, Management of MEC applications as VNFs, Compatibility with various VIMs and VNFM solutions
	LightMANO [57]	Compliance with ETSI MANO architecture, Orchestration of NFV services, Lightweight and scalable architecture
V2X Application	Veins [58]	Realistic simulation of vehicular mobility, Support for IEEE 802.11p (ITS-G5) and LTE, Evaluation of V2X communication protocols
	SUMO [59]	Realistic simulation of traffic scenarios, Time-discrete solution with adjustable step length, Coupling with ns3 for communication simulations [60]
Monitoring	Prometheus [61]	Metrics collection and storage, Dynamic service discovery, Alerting based on predefined thresholds
	Grafana [62]	Visualization of time-series data, Customizable dashboards, Integration with various data sources

tailored to address distinct requirements and functionalities within the dynamic MEC ecosystem:

(i) *ETSI MEC Sandbox* [63] provides an interactive environment for developers to experiment with standardized RESTful APIs tailored for MEC applications. It offers scenarios combining various network technologies and terminal types, allowing users to gain hands-on experience with APIs such as Location, Radio Network Information, and Traffic Management.

(ii) *Connected Vehicle Blueprint (CVB)* [64] offers a MEP with a focus on V2X, providing services to connected vehicles. These services are disseminated to applications hosted on vehicles based on predetermined policies for data dispatch and response. As the blueprint evolves, additional connected-vehicle applications and services are integrated. It provides MEC Components like MEP(s) and MEP Manager, supporting MEC APIs such as MEC 011, Mp1, and Mm5.

(iii) *Eclipse fog05* [65] provides a decentralized infrastructure for provisioning and managing compute, storage, communication, and I/O resources across the network. It caters to highly heterogeneous systems, enabling the deployment of MEC Applications on centralized or distributed MEPs. It offers an MEC Orchestrator and supports MEC APIs like the MEC 010-2 Application descriptor information model.

(iv) *Edge Gallery* [66] focuses on providing an open-source MEP framework at the carrier's network edge. It enables MEC edge resources and applications, providing security and management capabilities and offering interconnectivity with the public cloud. It aims to create a unified MEC application ecosystem compatible across a carrier's heterogeneous edge infrastructure. It provides the MEP and supports MEC APIs like MEC 010-2, MEC 011, and MEC 009, with plans for helping more ETSI APIs in future releases.

(v) *Eurecom MEP* [67] is a open-source implementation of MEP that provides discovery and registration of MEC applications and services. It provides supporting MEC APIs like MEC 011 Mp1, MEC 10-2, MEC 012 RNIs.

(vi) *Italtel's MEP (i-MEC)* [68] provides MEP and supports MEC APIs like MEC011 Mp1, Mm5 proprietary API, and Mp2 proprietary API (OpenFlow based).

(vii) *Serverless On Edge* [69] is based on a decentralized framework that includes the distribution of lambda functions across the serverless platforms. It offers a User app LCM proxy and supports MEC APIs like the MEC 016 Device application interface (Mx2).

(viii) *Simu5G* is a open-source solution not only supports simulating the data plane of a 5G network but also integrates an ETSI-compliant MEC system. Establishing RESTful APIs provides communication with MEC elements such as MEP and the User App LCM proxy. Simu5G supports some ETSI MEC APIs such as UE, Location, and RNIs [46].

According to ETSI ISG MEC, the MEC architecture only provides detailed specifications for some of its components. Some parts have clear instructions, especially for interoperability among different users. However, others may require proprietary solutions that must be fully detailed in the standard and are only explained at a functional level. Additionally, other organizations are developing some components, such as industry groups or open-source projects. The functional mapping of MEC activities to open-source solutions is depicted in Figure 2.7. As depicted, certain existing open-source solutions could potentially cater to specific MEC components. However, due to insufficient support and integration, implementing them remains challenging. Each open-source project tends to prioritize particular software elements. For example, some orchestration tools are designed for network automation management and do not fulfill the role of MEO as required in MEC environments [70].

While offering diverse functionalities and support for standardized APIs, listed MEC solutions exhibit limitations such as restricted use case coverage, integration complexities, and potential scalability issues. Additionally, varying degrees of API support, security risks, resource constraints, and deployment intricacies pose challenges for organizations. Concerns about vendor lock-in, lack of real-world testing, and performance overhead underscore the need for careful evaluation and consideration when implementing MEC solutions within edge computing environments.

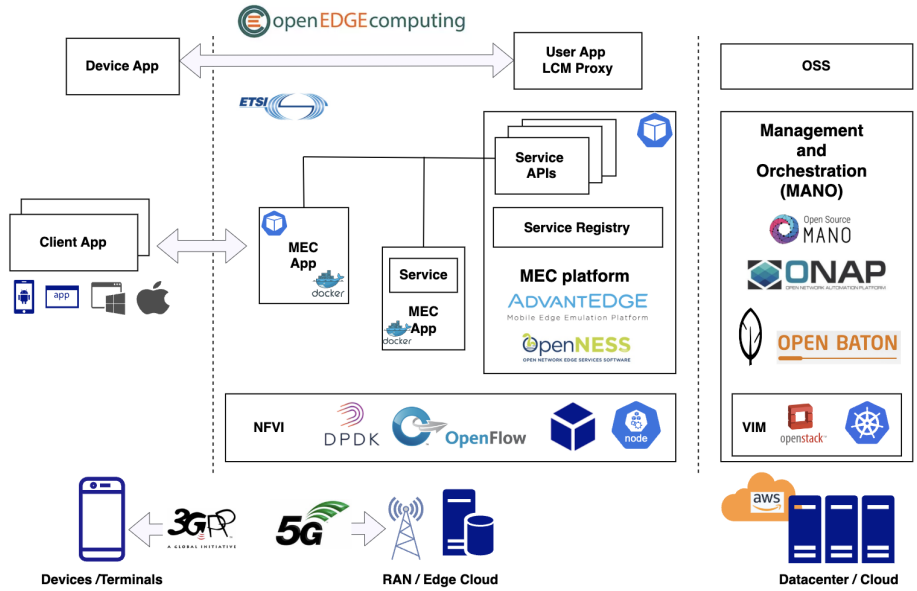


Figure 2.7: Functional mapping of MEC activities to open-source solutions.

2.4.2 AdvantEDGE - Motivation Behind the Choice

AdvantEDGE, operating on Docker and K8s, serves as a MEEP. It provides an emulation environment for experimenting with edge computing technologies, applications, and services. This platform facilitates the exploration of various edge deployment models and their impacts on applications and services through short and agile iterations [50]. The motivation behind using AdvantEDGE:

- Easy adoption of the ETSI-MEC framework and elements.
- Deployment and discovery of MEC applications and services as per the environment.
- Understand the deployment of MEH and MEC applications and their positioning in the network.
- Offer support for optimizing the network based on characteristics such as latency, jitter, and packet loss to understand their impact on applications and services.

- Analyze application behavior during user mobility within and across ANs.
- Built on K8s, enabling easy scalability as a VIM for managing multiple MEH instances.
- Incorporate relevant ETSI-MEC APIs, including location, radio network information, and V2X API for enhanced functionality.

AdvantEDGE utilizes a micro-service architecture to deploy edge applications in simulated network environments. It operates within Docker containers on K8s, with micro-services grouped into different categories. Key components include subsystems like platform, virtualization, access/admission, monitoring, metrics, sandbox elements such as traffic controller, mobility manager, and various compliant service APIs. The network model supports scenario definition with detailed characteristics like latency, jitter, throughput, and packet loss. AdvantEDGE can handle specific computing characteristics like the Central Processing Unit (CPU) and memory limits, with minimum and maximum values. Additionally, AdvantEDGE facilitates diverse edge application architectures, offering support for deployment, behavioral, and grouping models. It also enables experimentation with applications and services operating on nodes external to the platform [50].

2.4.3 Kubernetes (K8s) in MEC Framework

K8s is an open-source container orchestration system that streamlines application deployment, scaling, and management. Within cloud services, numerous providers offer K8s-based platforms or provide it as a service within their Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS) offerings. Many vendors also distribute their customized versions of K8s [50]. Figure 2.8 presents the K8s architecture that includes a cluster of the control plane (i.e., master node) and other nodes (i.e., worker node). Functionalities of the K8s components are divided into control plane and worker node.

The control plane functionalities are as follows:

- *Control plane* – The control plane component manages the cluster and makes global decisions about its state.
- *Kube API Server* – Exposes the K8s API and acts as the front-end for the K8s control plane.

- *Scheduler* – Assigns workloads (pods) to nodes based on resource availability and constraints.
- *Controller Manager* – Ensures that the cluster maintains the desired state defined in the control loop, managing aspects like node scaling and replication.
- *etcd* – A distributed key-value store used as K8s backing store for all cluster data.

The worker node is the machine in the cluster where containers are deployed, managed, and run. The worker node includes the following components:

- *Kubelet* – An agent running on each worker node ensures containers run in a pod.
- *Container Runtime* – The software responsible for running containers, such as Docker or containerd. Containerd is a lightweight runtime that manages the lifecycle and execution of containers used by platforms like K8s.
- *Kube-proxy* – Maintains network rules on nodes, enabling communication across the cluster and service abstraction.
- *Pod* – Pod is a K8s smallest deployable unit comprising one or more containers with shared resources.

In the context of MEC, K8s can function as a VIM, enabling efficient resource management and service orchestration at the network's edge. By utilizing K8s as a VIM in MEC, organizations can deploy and scale applications closer to end-users, thereby reducing latency and enhancing performance [72]. For instance, to simulate and deploy multiple MEH within a simulated environment generated using AdvantEDGE, K8s is an optimal choice as a VIM.

Compared to OpenStack, K8s offers several advantages within the MEC framework. Its lightweight and modular architecture makes it well-suited for dynamic edge environments, allowing for more effortless scalability and adaptability to changing workload requirements. K8s also provides robust automation capabilities, simplifying the deployment and management of containerized applications in MEC scenarios. Overall, K8s presents a more

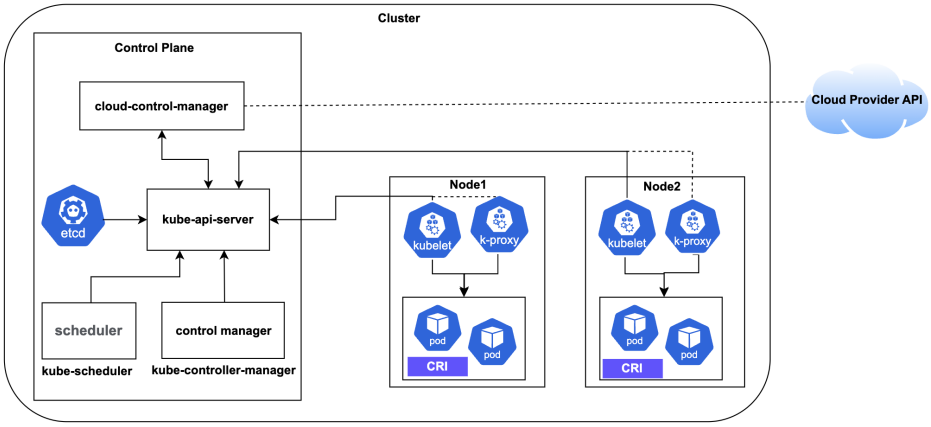


Figure 2.8: Kubernetes architecture [71].

agile and efficient solution for managing edge computing resources compared to traditional frameworks like OpenStack.

Moreover, K8s offers robust tools for managing and monitoring resource availability at MEC hosts, which is crucial for optimizing application deployment and performance. Through its comprehensive resource management features, K8s provides real-time insights into CPU and RAM usage at each MEC host. This is achieved via metrics collected from pods and nodes, which are continuously monitored and reported by the K8s control plane. By leveraging these real-time metrics, K8s can dynamically adjust pod deployment to maximize resource utilization and ensure optimal performance. For instance, K8s schedules pods based on current resource availability and demands and can perform automatic scaling of resources based on observed metrics. This dynamic adjustment helps in balancing workloads, preventing resource contention, and enhancing the overall efficiency of MEC-hosted applications. Additionally, K8s supports resource quotas and limits to enforce policies that ensure fair distribution of resources among applications, further contributing to effective resource management and load balancing in MEC environments.

In addition to resource management, K8s provides built-in load-balancing capabilities that effectively manage traffic distribution across MEC applications. Its load balancer ensures that traffic is evenly distributed among available pods, optimizing resource utilization and enhancing application performance. However, to achieve effective load balancing, it is crucial to

implement a strategy tailored to the specific needs of the MEC environment. This includes configuring appropriate load-balancing policies and continuously monitoring performance metrics to adapt to changing traffic patterns and workloads. By addressing these considerations, K8s can ensure that MEC applications perform optimally under varying conditions.

2.5 Orchestration of 5G-MEC Systems

Orchestrating 5G-MEC systems involves efficiently managing data and network resources to meet dynamic demands and enhance performance for low-latency, high-bandwidth services. Various strategies have emerged to enhance resource orchestration, including SDN and NFV. Resources such as computing, storage, and networking must be managed and coordinated to optimize specific tasks or host applications at the network's edge.

For instance, what can be orchestrated?

- MEH resources such as CPU and memory must be monitored and maintained efficiently. Additionally, the MEO may deploy MEC applications as per the demand of the system or end-user.
- MEC applications can be orchestrated to run at the network's edge depending on resource availability, latency requirements, and user proximity.
- NFs such as load balancing, content caching, security, and traffic optimization can be orchestrated at the edge to improve network performance and efficiency.
- Orchestration of data involves managing data storage and movement at the edge, including caching frequently accessed data and optimizing data transfers between edge and centralized data centers. Furthermore, user data must be protected and handled securely during the migration of a MEC application.

Orchestration decisions in MEC may based on the following factors:

- Service Level Agreements (SLAs) between service providers and consumers define criteria like latency, throughput, and availability, resource orchestration to meet service requirements.

- Decision-making regarding resource allocation at the edge dynamically considers the availability of computing, storage, and networking resources, allocating them based on demand.
- Orchestration factors in the location of users or devices accessing edge services, deploying edge applications closer to users to minimize latency and enhance user experience by reducing communication overhead.
- Orchestration prioritizes tasks based on computational intensity, data requirements, and sensitivity to latency, ensuring efficient execution of tasks at the edge.
- Strategies focus on optimizing costs by dynamically provisioning resources, their availability, and efficient resource utilization while minimizing operational expenses.

2.5.1 Resource Allocation in 5G-MEC System

In 5G MEC, resource allocation faces unique challenges due to the distributed nature of edge computing and the diverse requirements of applications. Critical analysis of resource allocation strategies must consider factors such as dynamic workload variations, heterogeneous resource availability, and stringent latency requirements. Various approaches, including heuristic algorithms, ML models, and game theory, are utilized for resource allocation in 5G MEC.

Building upon the challenges highlighted, researchers have delved into categorizing resource allocation problems in 5G MEC. The authors in [73] offer an in-depth analysis of how to categorize resource allocation problems in 5G MEC. This analysis takes into account various aspects like the goal of the allocation, the type of resources involved, the specific challenges encountered, and the underlying assumptions. Moreover, it introduces a systematic framework for classifying these issues, tailored specifically to tackle the diverse resource allocation challenges within MEC environments. This framework provides a structured approach to problem-solving, addressing different resource-related issues with precision.

Furthermore, the significance of factors such as security and dependability in resource allocation within networking and computing systems is paramount. The authors of [74] underscores the importance of considering

these factors when allocating resources, emphasizing their impact on the overall performance and reliability of the system.

5G resource allocation is a crucial aspect of managing network efficiency and performance. It involves dynamically distributing bandwidth, spectrum, and computing resources to ensure optimal connectivity for users and devices. In the resource allocation for 5G, the author of [75] underscores how network slicing facilitates the effective distribution of resources by establishing virtual networks over a common physical infrastructure, addressing varied requirements, including those of Mobile Virtual Network Operators (MVNOs). These slices encompass various resources like radio, CPU, memory, and bandwidth allocated to tenants or MVNOs for dedicated service delivery. Innovative resource allocation and pricing models, such as bidding schemes and market mechanisms, ensure fair and efficient distribution, maximizing network utilization while meeting user preferences and budget constraints. Additionally, strategic game-theoretic approaches help in determining resource pricing and allocation, fostering competition and optimization in the communication marketplace.

Finally, the author of [76] proposes a comprehensive resource allocation model for virtualized 5G networks in heterogeneous cloud infrastructures. It addresses the diverse requirements of network slices by considering resource demand vectors for each function within each slice. The model optimizes resource allocation to maximize overall network utility through convex optimization and introduces a distributed solution via resource auctions.

2.5.2 Implementation of the Orchestrator

The implementation of a MEC orchestrator involves configuring and customizing existing solutions to suit the requirements of a specific edge computing infrastructure [77]. The orchestrator must be capable of dynamically provisioning resources based on the requirements of MEC applications and services. This involves monitoring the status of available resources, such as processing power, memory, and network bandwidth, and making decisions in real-time to efficiently allocate these resources. Additionally, the orchestrator needs to support various orchestration policies and algorithms to optimize resource utilization and meet QoS requirements [78]. These policies may include load balancing, task migration, and fault tolerance mechanisms to ensure reliable and responsive operation in dynamic edge environments. Orchestration can be done in various ways, like arranging software and hardware services across different setups or organizing the

systems that support services, which can be made up of real or virtual computers, storage, and networks. OSM offers service orchestration and life-cycle management in more simple manner than the ONAP solution, but involves complex integration with existing VIM solutions such as OpenStack and Kubernetes [79].

In the intricate ecosystem of MEC, K8s [53] emerges as a cornerstone technology for containerized MEC application orchestration, offering a robust and scalable platform for managing containerized workloads across edge environments. The significance of K8s in MEC orchestration stems from several key attributes. First, K8s provides powerful primitives for deploying, scaling, and managing containerized MEC applications, abstracting the complexities of infrastructure management [71]. This enables seamless deployment of MEC applications across a distributed edge infrastructure, ensuring consistency and reliability. Second, K8s offers sophisticated resource management capabilities, allowing MEC applications to scale dynamically based on demand while efficiently utilizing edge resources. This adaptive resource allocation ensures optimal performance and cost-effectiveness in edge deployments. Third, K8s provides built-in service discovery and load-balancing mechanisms, enabling MEC applications to discover and communicate seamlessly with neighboring services. This facilitates the creation of resilient and highly available MEC architectures capable of handling varying workloads and network conditions. Furthermore, K8s incorporates robust fault tolerance and self-healing mechanisms, automatically detecting and recovering from failures in MEC applications and infrastructure components [80].

Figure 2.9 represents one of the scenarios discussed in [18]; the deployment of the MEC structure may vary depending on the specific implementation and requirements of the network provider. Generally, each MEH may have one MEPM responsible for overseeing the MEP's operation on that host. However, in some scenarios, especially in large-scale deployments or when using virtualized or distributed architectures, multiple MEHs may share a single MEPM for efficiency and resource optimization. It ultimately depends on factors like network topology, resource availability, and management preferences of the MEC deployment.

Maintaining service continuity becomes crucial as MEC applications are exposed to UE mobility. Application mobility, facilitated by transferring application instances and user context between MEHs, ensures seamless service delivery despite UE movement. While stateful services require synchronization of application states, stateless services simplify the process.

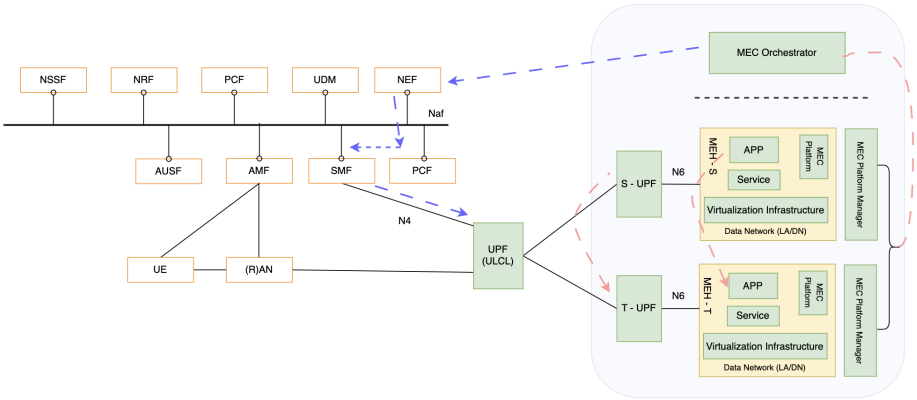


Figure 2.9: Interaction of MEC orchestrator.

ETSI ISG MEC is actively developing procedures for application mobility, which involve enabling, detecting, validating, and processing application relocation triggered by UE movement [81]. Detection often relies on 5G NEF and RNI [18]. The complexity of services provided by MEC applications necessitates careful consideration of the application life cycle, including mobility support, by service providers.

2.6 Vehicle-to-everything (V2X) Communication

The connected car market has received investments from both the automotive and telecommunication industries, recognizing its support for digital transformation by exploiting communication technologies [82]. Telco alliance Next Generation Mobile Networks (NGMN) backs Cellular-V2X (C-V2X) technology [83] for connected cars from a technical and economic perspective. V2I, V2P, and V2N interactions presented in V2X communication are crucial for improving road safety and traffic efficiency. Integrating MEC within the 5G ecosystem is actively supported to meet the low-latency requirements of V2I communications (targeting less than 10 ms [84]). These requirements make MEC a promising technology that meets demanding performance criteria. MEC streamlines direct node-to-node communication, enabling seamless exchange of V2X data over the underlying network infrastructure.

V2X communication encompasses a range of interactions between vehicles and their surroundings, pivotal in enabling Cooperative Automated Driving

Table 2.2: Performance requirements for V2X use cases.

UC No.	Title	Delay			Throughput		
		5GAA	3GPP (ms)	ETSI (ms)	5GAA	3GPP	ETSI (Mbps)
1	Platooning (Lowest Degree)	Delay Sensitive	10–25	25	8–48 kbps	-	
2	Remote Driving	Delay Sensitive	5	5	400 kbps–36 Mbps	1–20 Mbps	UL: 25 DL:1
3	Lane Change	-	10–25	10	120 kbps	-	
4	Collision Avoidance	-	10	10	10 Mbps	-	10
5	Intersection Crossing	-	-		8–25 kbps	50 Mbps	
6	Emergency Trajectory Alignment	-	3 ms	3	48 kbps	30 Mbps	30
7	Cooperative Driving (Highest Degree)	-	5 ms	20	-	384 kbps	65

(CAD) functions, ensuring safe and reliable distributed driving across communication partners [85]. The transition from LTE-based C-V2X to 5G New Radio (NR) technology promises higher throughput, lower latency, and increased reliability in dense traffic scenarios, highlighting the advancements for V2X applications [86].

To support V2X communication, several standardized bodies are working to identify critical issues and bring together requirements from vendors and network providers to mitigate interoperability issues. Furthermore, the proposed solution is enhancing, and more industries are getting involved. 3GPP, 5G Automotive Association (5GAA), and ETSI are among the significant collaborators. Both 3GPP and 5GAA are working on specifying V2X communication over 5G [16]. ETSI is standardizing MEC responsibilities and supporting V2X applications in various scenarios [15].

Table 2.2 presents the performance requirements i.e. latency and throughput for various V2X use cases provided by 3GPP, 5GAA and ETSI [16, 87, 88].

2.6.1 MEC in Vehicular Networks

MEC brings computational power at the network's edge closer to end-users such as vehicles and pedestrians. ETSI-MEC provides support to the V2X use case and is categorically divided into three groups: (i) safety, (ii) convenience, and (iii) advanced driving assistance [89]. Safety encompasses various MEC-relevant use cases supporting road safety via V2I and V2V communication. Software updates and telematics, typically part of the convenience group, are feasible with current access technology and partly funded by car makers. VRU use case involves pedestrians and cyclists, requiring accurate positioning data for effective information use. Furthermore, ETSI-MEC analyzes critical issues such as user mobility and QoE

support, maintaining low latency within a multi-operator scenario, and ensuring reliable communication with vehicles to coordinate traffic.

ETSI MEC provides VIS API [28] that exchanges relevant V2X data necessary as per the demand of the environment. VIS is designed to support V2X interoperability within a diverse environment encompassing multiple vendors, networks, and multi-access environments. VIS API discusses multi-operator scenarios, application initiation, and impact on user mobility. Whereas [16] reports the performance requirements for use cases of vehicle platooning, advanced driving, extended sensors, and remote driving. UE supporting V2X applications and the V2X application server necessitate a data rate of 10Mbps for video sharing in the uplink mode [16].

ETSI MEC categorizes several use cases and scenarios that may require computational capabilities at the network's edge. In TR 22.886, one of the use cases specifies that a UE can be equipped with a camera to process the video of the environment and send it to the nearest MEH. MEH location is not discrete; MEH may be placed at the network's edge or in the cloud, depending on the requirements. Furthermore, MEH will post-process the video depending on the computational task and responsibilities [90]. In addition, ETSI categorizes the solutions based on 1D, 2D, and 3D objects. 1D objects include traffic lanes, 2D includes the traffic sign whereas 3D object includes pedestrian, vehicles [90].

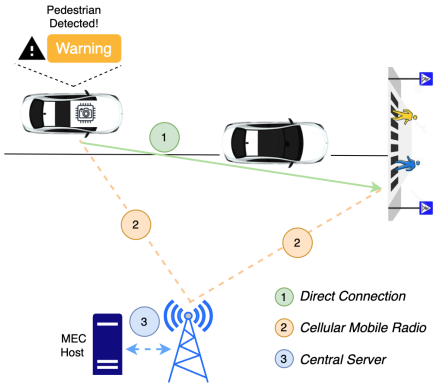


Figure 2.10: General V2X use case concept in MEC.

Figure 2.10 presents a specific V2X use case with MEC support [91]; the work focuses on the VRU detection and collision avoidance system while using MEC. 5GAA also presents a VRU use case, selected for phase 1 of

the MEC trials in Europe (EU) and North America (NA), which involves a cooperative awareness-based approach where location and dynamics data are exchanged between host vehicles (HV) and VRUs, facilitated by ML applications in the edge cloud. MEC performs an essential role in analyzing trajectories, predicting collisions, and alerting vehicles by leveraging infrastructure sensor inputs and vehicles awareness data [92].

2.7 Overview of the 5G-MEC Ecosystem for V2X Applications

Figure 2.11 depicts a 5G-MEC ecosystem with distributed elements for V2X applications, the focus of this thesis. These elements cover both hardware and software functions of 5G-MEC for V2X applications. Hardware components include Vehicular User Equipment (VUE), Road Side Unit (RSU), and RAN infrastructure, while gNB (Next Generation NodeB) and eNB (evolved NodeB) serve as physical components in 5G and 4G networks, respectively. The software elements are distributed and present in every hardware component, whereas MEH can be deployed within the RAN infrastructure. Typically, MEH is deployed in a VI and hosts MEC Application (MEC App). 5G CN elements include: UPF, responsible for routing and processing data packets; AMF, which manages access to the 5G network and handles mobility; SMF, controlling the establishment, modification, and termination of data sessions for UE; NEF, the primary entity facilitating the exposure of network capabilities and services to edge applications; and orchestrator, which provides centralized management and coordination of various components and resources. The reference points represent the connectivity and data traffic flow between various elements of the 5G-MEC system for V2X applications. The placement of these elements may vary depending on the scenarios and network provider.

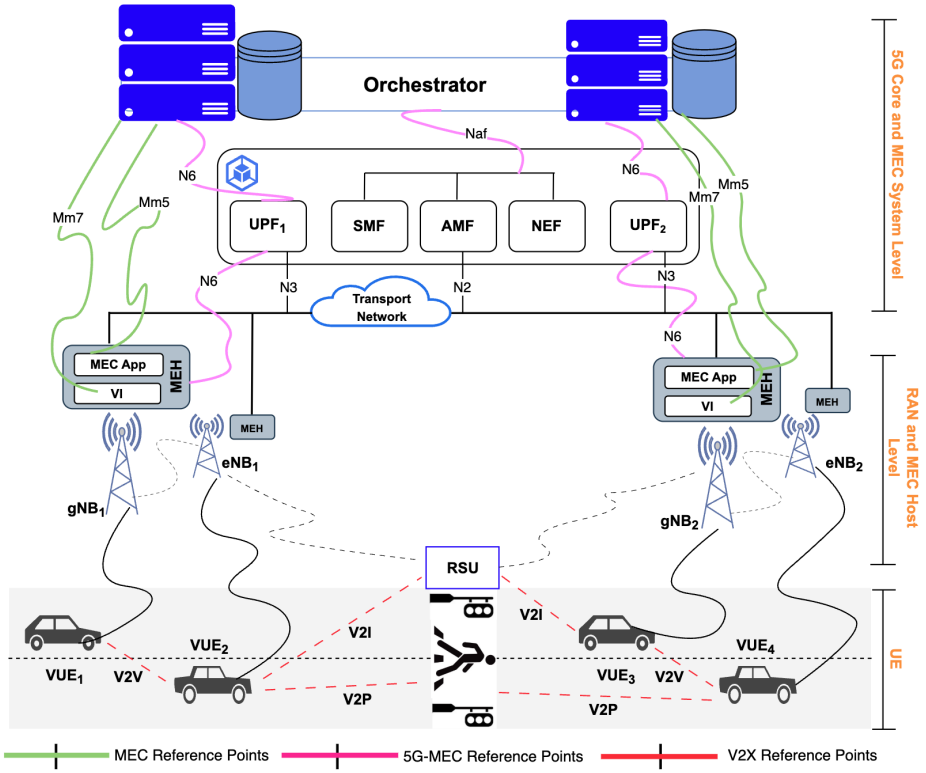


Figure 2.11: 5G-MEC ecosystem for V2X applications.

Chapter 3

Related Works

This chapter offers an overview of the current state-of-the-art 5G-MEC testbeds alongside relevant studies that emphasize specific experimental investigations conducted within these testbed environments and the scope of the thesis. Initially, a survey and existing studies are provided based on 5G-MEC testbeds for V2X applications. Later, the focus is on orchestrating MEC applications using different strategies. Furthermore, optimization algorithms are analyzed to allocate resources and offload tasks in MEC environments.

3.1 State-of-the-art of 5G-MEC Testbeds for V2X

To implement 5G-MEC testbeds, industry, and standardization bodies have established several notable testbed deployments and experimental frameworks. These platforms validate the feasibility and performance of 5G-MEC-enabled V2X solutions in real-world scenarios. A comparative analysis of these initiatives identifies common architectural patterns, deployment considerations, and performance metrics informing the design and implementation of effective V2X testbeds. Within the V2X use case, multiple scenarios can be considered based on the specific categorical use case. Moreover, incorporating 5G and V2X components can encompass simulated, emulated, or practical setups, facilitating the support for real-world MEC deployment. This research contributes to comprehending the integration of software elements and open-source software packages for constructing these components.

The authors of [93] offer a comprehensive review of small-scale testbeds for network slice implementation, highlighting the lack of similar studies in this domain. The paper identifies critical enabling technologies and maps open-source software packages to the ETSI NFV MANO framework while defining primary and secondary design criteria for deploying such testbeds. Additionally, it presents leading small-scale testbeds and evaluates them against the predefined criteria, concluding with a discussion on common deployment challenges. The authors of [94] present various use cases, applications, and integration challenges concerning 5G and MEC. Additionally, it discusses studies and testbeds focusing on utilizing single-board computers for edge clouds, lightweight platform implementation, and middleware for edge computing. The authors of [95] offer a survey that concentrates on provisioning MEC resources, addressing the allocation of MEC resources, deployment bottlenecks, and associated costs within a smart metropolitan area, fostering the coexistence of diverse verticals. The paper examines various open research challenges highlighted in the surveyed study and their corresponding use cases. The authors of [96] explore existing MEC initiatives, their strategies, limitations, impact, deployment efforts, and tools, aiding in designing and improving MEC systems.

3.1.1 Survey of Current 5G-MEC Testbeds

Several existing testbeds and implementations for 5G-MEC are available with different use cases and scenarios. 5GCity testbed [97], spanning Bristol, Lucca, and Barcelona, features diverse platforms and services catering to applications like video streaming and smart cities. Its MEC-neutral host platform, driven by OpenStack and SDN RAN controller, enables scalable orchestration. Additionally, utilizing OSM, it facilitates seamless creation and deletion of MEC applications via proprietary Mm1 interface, supported by MEPM functionalities and Mp1 interface interaction for MEC applications, complemented by a comprehensive Software Development Kit (SDK) for resource placement, network slice application, and monitoring [98].

Mosaic5G platform [99], with its flexible and scalable service deployment, integrates five software elements, including OAI for LTE network functionalities [41]. FlexRAN [100], an open-source SD-RAN implementation, enables centralized or scattered control policies across base stations, enhancing task management within the RAN. LL-MEC segregates data and controls plane traffic at the network edge, empowering MEC functionalities and

services, while the NFV MANO platform using JOX ensures end-to-end network slice provisioning, supported by monitoring and application control modules within the Store component.

University of Bristol's testbed [101] tailored for innovative city applications, combining open source and licensed solutions. CloudBand by Nokia hosts MEP and MEPM, executing MEC functions alongside VIM, while NetOS drives the SDN architecture and OSM orchestrates VNFs. InterDigital furnishes media services, like content distribution, vital for the 5G smart tourism initiative, managed via OpenStack, with mini-scale sensor nodes and Raspberry Pi equipment gathering and processing essential data at the data center.

Linux Foundation Edge (LF Edge) project [48], establishes an open source MEC framework for Industrial Internet of Things (IIoT) applications at the network's edge, with its Golang version, EdgeX-Go, featuring seven subdirectories including command(cmd) for project program entry, APIs for microservices, Docker for image creation instructions, and Internal for device microservice initialization. This framework enables data conversion from diverse sensors and devices into common structures, alongside customer-specific data provision via Transmission Control Protocol/Internet Protocol (TCP/IP) protocol, catering to applications, enterprises, and cloud systems, capable of running on small-scale equipment like Raspberry Pi [102].

The authors of [103] introduce a 5G MEC testbed based on OAI featuring VI and showcased its capabilities through a video streaming application with object detection scenarios supported by network slicing. The authors of [104] improve a 5G emulation framework by integrating ETSI MEC location service API and exhibited proximity alerts for drivers using the SUMO platform and app. The authors of [105] propose an innovative ETSI MEC-compliant architecture tailored for advanced 5G deployment environments, validating its efficacy and adaptability through simulation and experimental tests within a 5G network context. The authors of [106] developed a simulation model of the proposed ETSI MEC extension using the Objective Modular Network Testbed (OMNeT) simulation tool and demonstrated its applicability in future (beyond) 5G scenarios. The authors of [107] systematic survey and categorization of open-source testbeds utilizing commercial-off-the-shelf hardware for integrating Time-Sensitive Networking (TSN) with 5G networks. Through measurement verification on their in-built TSN testbed, the paper provides insights into the baseline performance of 5G-TSN integration and suggests open research directions for further development in this field.

VITAL-5G [108, 109] facilitates experimentation and validation of Transport & Logistics (T&L) services within real-life 5G environments like sea ports, river ports, and warehouses through its accessible platform. The platform provides an advanced 5G-enabled experimentation facility with EdgeApps, enabling T&L developers to test and validate applications effectively. The authors of [110], showcases the Port of Antwerp testbed, illustrating how VITAL-5G connects to diverse 5G testbeds and deploys vertical services, emphasizing EdgeApp MANO. [111] proposes a solution for dynamically managing network slices in 5G ecosystems, mainly focusing on the T&L sector. The paper introduces a slice orchestration system that interacts with testbeds, monitoring performance and adjusting slices to meet specific requirements such as latency and throughput. The authors of [112] introduce a virtualized evaluation testbed using open-source software components to support key 5G architectural concepts like NFV and MEC. The paper addresses performance and versatility standards by offering a customizable platform with container and VM orchestration aligned with NFV and MEC principles.

In addressing the complex integration of 5G private networks into smart factories, the 5G CONNI project tackles challenges related to network architecture, operator models, and the management of both public and private elements [113]. The project focused on four main implementation stages: defining the operator model, understanding industrial application requirements, transforming VNFs, and establishing monitoring and backup mechanisms. The proposed ECoreCloud platform explores relocating physical machine services to the cloud for greater flexibility, while maintaining response times under 30 ms.

The authors of [114] of addresses the problem of efficiently implementing transparent access to 5G edge services using SDN to ensure low latency and scalability. It proposes a modular architecture with multiple filter stages to minimize unnecessary traffic to the SDN controller and optimize flow table usage in hardware switches. The authors conclude that their solution improves performance and scalability, as demonstrated through a performance evaluation on a real edge/fog testbed.

Analyzing the KPIs of the above-mentioned testbeds and implementations provides valuable insights into their performance, scalability, and suitability for different applications and scenarios within the 5G-MEC testbed.

3.1.2 5G-MEC Testbeds: V2X Applications

5GVINNI Munich testbed [115] serves the eHealth use case "Remote interventional support for emergency care application—mobile ultrasound." It integrates MEC functionalities, 5G RAN, and Huawei's 5G core, employing a floodlight controller for SDN and Huawei's MANO and NFVI. Docker containers and Mininet facilitate NFVI deployment, enabling SDN capabilities for NF inter-connectivity via OpenvSwitch. Ambulances with 5G modems connect to the 5G site via an ethernet link. At the same time, MEC services and applications are deployed using Docker within the MANO framework, allowing VNF instantiation and termination managed by VNFM [116].

5G-VINNI moving experimentation facility [117] for public safety and disaster relief, housing the SATis5 testbed [118] within a Rapid Response Vehicle (RRV). SATis5 integrates an edge node and central core network node for satellite backhaul, enabling UE connection via the same network. Fraunhofer FOKUS Open5GCore toolkit implements 3GPP 5G core network functionality, supporting NSA architecture with 4G eNB for RAN due to the unavailability of commercial 5G gNB, integrating with iDirect satellite hub platform for network operations virtualization via OpenStack Pike VIM.

Smart Highway V2X testbed [119], situated in Antwerp (Belgium) as part of CityLab, features RSUs facilitating V2X radio connections such as Intelligent Transportation System-G5 (ITS-G5) and C-V2X. Experimenters can select communication modules to test on RSUs that are fiber-connected to a virtualized cloud-based backbone equipped with commercial and Software-Defined Radio (SDR) modules. The testbed integrates MEC collocation with RSUs, deploying K8s clusters for monitoring services, supporting MEC applications, and enabling real-time decision-making by the containerized MEAO (cMEAO) [61].

5G Carmen testbed [120] integrates diverse MEC systems across Italy, Germany, and Austria, featuring interfaces for Packet Gateway connection, public IP exposure, and VPN for management functions and third-party applications. In Germany, Deutsche Telekom collaborates on LTE/5G MEC, leveraging Nokia infrastructure for edge processing and latency-optimized access, utilizing Airframe hardware and Ubuntu Linux/KVM virtualization. In Italy, TIM partners with Nokia for MEC solutions, employing container-based service mechanisms, AMQP broker platform, and VM orchestration via Nokia CBAM, while Austria's setup, overseen

by Magenta Telekom Austria (MTA), adopts OpenStack Red Hat Release 13 for hosting MobiledgeX's low-latency computing platform.

5G-DRIVE testbed [121], centered on eMBB and V2X scenarios, employs Nokia's NetLeap LTE test network infrastructure, featuring a virtual mobile network with its EPC. Interconnecting eNodeBs with OpenStack cloud and SDN-enabled backhaul, experimental tests were conducted at a public site in Finland, focusing on ITS-G5/LTE-V2X coexistence. Equipped with a C-V2X box, traffic light with C-V2X RSU, and car tablet, the field test assessed latency and packet loss rate between RSU and OBU, utilizing LTE network (2.6 GHz), 5.9 GHz RSU, and NEBULA OBU [122].

The authors of [123] discussed the progress made in autonomous driving technology across various levels of automation, from Level 1 to Level 5. The paper has developed a framework for enhancing autonomous driving capabilities through Connected, Cooperative, and Automated Mobility (CCAM) infrastructure, specifying technical requirements and validating them using ML algorithms to predict traffic flow and mobile communication demands. The authors of [124] present the orchestrated edges platform for managing distributed edges in 5G environments, catering to CCAM use cases. The paper ensures service continuity for vehicles across borders, validated through the 5G-CARMEN trials [120], leveraging a reference architecture integrating 5G system components and MEC facilities for dynamic deployment and management of edge services.

Above mentioned testbeds in Subsections 3.1.1 and 3.1.2 depict dedicated use cases, addressing specific requirements such as MEC functionalities, and V2X communication. However, the overall implementation of a testbed lacks standardized protocols and interoperability frameworks and its usage complexity. Prior surveys [93, 95, 96] have detailed specific attributes like network slicing, resource allocation, and hardware and software solutions tailored to each layer of a 5G-MEC systems. This motivates the thesis's contribution to scrutinize the current array of solutions accessible for 5G-MEC in V2X applications, exploring the implementation of such testbeds, assessing their potential usability, and determining strategies for replication.

3.2 Orchestration in 5G-MEC Testbeds

Orchestrating MEC applications and MEHs is crucial for maintaining seamless connectivity and high performance in modern communication systems. Various studies address a spectrum of challenges and propose

advanced solutions for optimizing the management of MEC services, ensuring QoS and QoE, and enhancing network flexibility and reliability. By leveraging cutting-edge technologies such as SDN, and NFV these studies explore innovative strategies for orchestrating MEC environments, including mobility-aware computation offloading, dynamic spectrum sharing, and real-time automation. The integration of modular frameworks, programmable interfaces, and automated deployment tools further underscores the ongoing efforts to create robust, scalable, and adaptable MEC infrastructures capable of supporting the complex demands of next-generation applications.

The authors of [125] focus on the orchestration of individual services and the network of MEPs supporting mobility. The paper advocates for a more sophisticated service and resource management framework integrating networking and cloud orchestration.

The authors of [126] point out the challenges posed by the heterogeneity of services, resources, technologies, and cloud infrastructure in maintaining service continuity and ensuring QoS and QoE. The paper provides mobility-aware computation offloading, dynamic spectrum sharing, and interference mitigation, leveraging technologies like SDN and NFV to ensure responsiveness, reliability, and resiliency. It stresses the importance of leveraging NFV, SDN, and MEC to create a customizable architecture tailored for CCAM applications. Solutions include mobility-aware computation offloading, dynamic spectrum sharing, and interference mitigation, leveraging technologies like SDN and NFV to ensure responsiveness, reliability, and resiliency.

The authors of [127] introduce a modular programmable MANO framework and a Service Construction Kit (SCK) to enhance NFV DevOps cycles. The MANO framework enables customization per service or VNF, while the SCK aids developers in local testing and interfacing with operational NFVI, improving flexibility and reducing development time. They suggest constructing function- and service-specific managers described and configured within VNF Descriptors (VNFDs) and Network Service Descriptors (NSDs) to support customized MANO.

Several studies delve into different aspects of network service orchestration. The authors of [128] distinguish key concepts of network service orchestration and provide a taxonomy of orchestration approaches, laying the groundwork for various orchestration scenarios. They highlight the need for a Multi-Domain Orchestrator (MDO) to coordinate resources and services across multiple administrative domains and technologies. Moreover, it

reviews existing solutions for NSO within evolving network service provider business models, focusing on coordination across diverse infrastructures and technologies like 5G, SDN, and NFV.

The authors of [129] propose a solution for orchestrating network monitoring services in SDN-oriented networks, focusing on real-time automation. The paper defines an architecture integrating SDN and NFV technologies to manage the lifecycle of monitoring services by considering control plane information. This paper explores incorporating monitoring into the ETSI NFV architecture for 5G, focusing on monitoring control and data planes separately.

The authors of [130] showcase four modes of MEC server selection strategies for optimizing communication and computational latency in a traffic management scenario. The paper involves migrating MEC servers based on distance, load, or both, demonstrating differences in ambulance response time and total drive time. The implementation utilizes Unity 3D for car simulation, data collectors for aggregation, analytical entities for coordination, and a MEC orchestrator for server selection, orchestrated via K8s and encapsulated in Docker containers.

The authors of [131] introduce a test platform prototype for emulating Points of Presence (PoPs) to test MANO systems, focusing on large-scale testing of OSM versions. The paper presents a platform capable of emulating multiple NFVI environments on a single machine, significantly reducing setup time and enabling pre-validation of MANO systems for large-scale 5G scenarios.

The authors of [132] address the challenge of testing and validating NFV MANO systems by proposing MANO-specific KPIs and comparing the performance of ONAP and OSM projects. The paper highlights the complexities and gaps in current open-source MANO projects, emphasizing the need for standardized performance benchmarks. The analysis reveals shortcomings in runtime orchestration actions, indicating challenges in assessing the reliability and quality of deployment for both platforms.

MEC benefits from SDN and NFV by enabling software-based MANO. For instance, the authors of [133] provide an architecture that combines SDN and MEC technologies to orchestrate a vehicular network, ensuring reliable communication with low latency for connected vehicles. The paper integrates various access technologies, including IEEE 802.11p and 5G, and utilizes MEC to reduce overall delay by offloading traffic from the backbone network. Through a practical use case, the architecture demonstrates

scalability, responsiveness, and reliability, with SDN facilitating flexible hardware deployment and MEC enhancing user experience with cloud-based services.

The authors of [134] survey MEC architecture, particularly automated deployment, by using Infrastructure as Code (IaC) tools, which are crucial for maintaining optimal policies in dynamic edge environments. IaC tools enable automated and consistent deployment, which aligns with the need for efficient management and orchestration of edge computing resources.

3.2.1 Kubernetes (K8s) for MEC

As edge environments become increasingly complex, leveraging K8s has become pivotal for managing containerized applications within MEC systems. K8s in MEC environments can be leveraged for dynamic MANO of containerized applications, facilitating seamless scaling, efficient resource allocation, and robust deployment of edge services. Several initiatives are underway to deploy and advance K8s in alignment with the functionalities delineated by the ETSI-MEC framework.

The authors of [135] provide K8s integration into MEC architecture to replace virtualized applications with containerized instances, enhancing scalability and load balancing while ensuring standardized communication interfaces. This approach, aligned with ETSI standards, simplifies adoption for telecom providers and enables the creation of scalable MEC infrastructure using federated K8s clusters.

The authors of [136] address the challenge of deploying, scaling, and managing container-based applications in centralized data centers and edge environments, exacerbated by the development of 5G technology. They propose a common platform utilizing K8s to seamlessly handle these tasks, achieving high availability, rapid deployment, and upgrade times. They demonstrate the platform's effectiveness in meeting cloud and edge 5G application requirements.

The authors of [137] conducted a study based on container-based service deployment, establishing a benchmark for MANO solutions in the MEC context. The authors of [138] propose the MARSAL MEC framework, leveraging a subset of ETSI MEC/NFV where K8s assume the role of VIM. The authors of [139] demonstrate a novel slicing architecture where orchestration and slicing for MEC applications benefit from using K8s and Helm technologies.

The authors of [139] propose an architecture integrating K8s into MEC environments for dynamic MANO of containerized applications, enhancing end-to-end latency guarantees in 5G network slicing. The paper introduces concepts like Application Slice (APS) and Application Service (AS) to distinguish between application and network services. It proposes a multi-tenant MEC architecture compatible with 3GPP standards to support performance isolation and efficient resource sharing between different slices.

K8s is employed in MEC [140] for dynamic MANO of containerized applications, facilitating efficient resource allocation and scaling at the edge. Through analysis of latency, power consumption, and responsiveness, K8s demonstrates its capability to enhance resource efficiency and scalability in MEC environments for real-time industrial monitoring and automation.

The authors of [141] develop a framework for deploying and managing IoT applications in MEC environments by using K8s as the orchestration and management tool. This framework leverages microservices architecture and containerization technology to ensure stability, scalability, and efficient resource utilization.

3.2.2 Migration Strategies of MEC Applications

The relocation of MEC services is a critical aspect of application management, necessary for managing server loads and accommodating client mobility. The authors of [96] discuss mobility-related issues, primarily focusing on the optimal instance for migrating MEC applications and which content to migrate to enhance QoE. Different mobility factors are considered. The authors of [142] present the optimal approach to migrating MEC applications and complete migration strategies to reduce energy consumption. The authors of [143] consider a prototype system approach at the network layer to manage seamless connections between edge servers and mobile devices. Some works conduct experimental tests using different MEC models and migration strategies, with one presenting K8s as the MEO [144]. This work proposes reactive service migration with the Evolved Packet Core (EPC). Other experimental studies integrate OSM, an orchestrator, with OpenNESS [145], a MEP, to migrate the MEC applications between MEHs [146]. The study includes two components, one maintaining the application's state with the client and the other focusing on management.

The authors of [147] discuss various container and migration strategies, focusing on the fog, edge, and cloud. The work emphasizes current approaches and frameworks for container-based service migration. In [148],

the authors describe different methods for migrating pods in K8s and present results on downtime with and without migration, along with the data transfer size. A prototype approach using an extended version of kubelet and customized containers is available on GitHub [149] for stateful container migration. This prototype approach extends the kubectl command to include a command for checkpointing and migrating running pods in K8s. The prototype implementation consists of a pod migration operator with custom resources and a controller at the control plane.

MEC application migration can be achieved using pod migration in K8s due to the shared underlying architecture between the two. There are several existing studies on efficiently migrating pods. Pod migration involves relocating containers, specifically Pods within K8s clusters, from one location to another without disrupting the services they deliver[150]. This migration is particularly critical in geo-distributed environments or fog computing infrastructures. Techniques like checkpoint operations, capturing state parameters, and transparent snapshotting of disk states are utilized to ensure smooth migration [151]. Systems such as MyceDrive [152] have been developed to facilitate stateful resource migration within K8s orchestrators, significantly reducing downtimes.

As the significance of orchestration in MEC becomes increasingly apparent, its role in facilitating various scenarios and addressing associated challenges is evident in Section 3.2. Whereas several studies in Subsection 3.2.1 highlight the advantages of employing the K8s framework within MEC environments. The complexities outlined in the above-mentioned work underscore the importance of delving into orchestration strategies for deploying MEC applications/services within the 5G-MEC infrastructure. This motivates the necessity of exploring how orchestration frameworks can adeptly manage and synchronize the varied components, encompassing services, resources, and technologies. This is crucial to guarantee the smooth deployment and service continuity of MEC applications/services in the dynamic landscape of 5G environments and user mobility.

The authors of [153] presents an edge cloud infrastructure testbed designed to study mobility scenarios in MEC. It highlights the flexible and scalable nature of the testbed, which uses commodity hardware and K8s for managing edge devices. Initial experiments focus on task migration due to edge device overload and unpredictable user movements. The results demonstrate the feasibility of task migration and offer insights into system performance under various conditions. While the testbed effectively demonstrates task migration capabilities and system performance, it primarily

relies on simulations and may not fully capture real-world complexities, such as diverse network conditions and varying user behaviors.

Within the V2X framework, orchestrating and migrating MEC services are crucial for ensuring seamless and efficient communication between vehicles, infrastructure, pedestrians, and networks. Orchestrating services within containers based on analyzed processes and managing the migration of virtual machine volumes between software-defined storage systems are vital for uninterrupted operations. Together, these actions enable reliable, real-time, high-performance V2X communications, enhancing the overall functionality and user experience of intelligent transportation systems.

3.3 Multi-Objective Resource Allocation in 5G-MEC Testbeds

In 5G-MEC systems, multi-objective resource allocation refers to distributing resources across various tasks or applications while simultaneously considering multiple objectives. Optimizing resource allocation across multiple objectives aims to enhance overall system performance, scalability, and user satisfaction while ensuring efficient resource utilization and cost-effectiveness.

The authors of [154] propose a Multi-objective Resource Allocation Method (MRAM) for IoT applications in MEC that addresses challenges related to meeting service requirements such as completion time, load balance, and energy consumption. MRAM utilizes the Pareto Archived Evolution Strategy (PAES) to optimize these objectives and employs the technique for order preference by similarity to ideal solution (TOPSIS) and Multiple Criteria Decision-Making (MCDM) to determine the optimal resource allocation strategy.

The authors of [155] tackle wireless resource sharing in MEC-caching-coexist (MCCe) systems, employing a multi-objective approach to simultaneously minimize transmission delay and energy consumption. They introduce an MODRL/HA algorithm, which combines Envelope Updated Design (EUD) and Parameterized Network Design (PND) techniques to handle multi-objective optimization and hybrid actions, achieving a 22% improvement over benchmark schemes in simulations.

The authors of [156] address the challenge of jointly optimizing task offloading, power assignment, and resource allocation to maximize user

offloading benefits while minimizing response time, energy consumption, and cost. The proposed algorithm, a multi-objective evolutionary algorithm based on decomposition (MOEAD_MEC), effectively balances these conflicting objectives through a multi-objective evolutionary approach, leading to significant improvements in user offloading benefits, as validated by comprehensive simulation experiments.

The authors of [157] optimize the resource allocation in MEC systems by offloading tasks from mobile devices to edge servers. They propose a multi-objective hybrid accelerated particle swarm optimization and dynamic program (MOAPSO-DP), which uses advanced algorithms to minimize computing time, service cost, and waste while maximizing task associativity with edge servers.

The authors of [158] address a multi-objective optimization for Service Function Chains (SFCs) placement in 5G-MEC environments, considering factors like end-to-end latency, resource congestion, and service acceptance ratio. It implements a novel Deep Reinforcement Learning (DRL) algorithm, the Chebyshev-assisted Actor-Critic SFCs Placement Algorithm, to efficiently allocate resources, achieve high service acceptance ratios, and avoid congestion under varying workload scenarios.

The authors of [159] propose dynamic optimization schemes for computation offloading and resource allocation in 5G MEC heterogeneous networks. The paper addresses the challenges posed by dynamic computation request arrival, energy availability, radio network conditions, and computation resources at MEC servers by developing integrated schemes for static and dynamic subchannels during time slots.

Furthermore, the relevant works on multi-objective approaches for specific scenarios such as migration are reviewed and categorized below based on whether the focus is algorithmic or experimental.

3.3.1 Algorithm-based

The authors of [160] focus on addressing the task migration problem in MEC systems caused by distributed user mobility, aiming to minimize the average completion time of tasks under a migration energy budget constraint. The paper employs a reinforcement learning algorithm based on a Markov chain model, considering the memoryless movement of users. Based on counterfactual multi-agent (COMA) reinforcement learning, the proposed distributed task migration algorithm facilitates user cooperation to optimize task migration while considering energy constraints.

The authors of [161] implement dynamic traffic steering in a 5G-enabled MEC framework, focusing on seamless service migration. The paper introduces a distributed approach and novel algorithms to optimize path selection based on time delay and available bandwidth. Experimental validation shows significant improvements in QoS efficiency, addressing challenges like live service migration and conflicting bandwidth requirements.

The authors of [162] focus on optimizing response time in mobile edge-cloud computing systems by deploying heterogeneous edge servers. The paper addresses the neglect of server heterogeneity and response time fairness in existing schemes. The proposed approach, utilizing offline Integer Linear Programming (ILP) for server placement and online game-theory-based methods for dynamic user movement, significantly reduces system response time by 47.37% and improves response time fairness by 71.60%, as demonstrated through extensive experiments.

The authors of [163] propose an ML-based server deployment policy for Edge Cloud Computing (ECC) in 6G IoT environments, aiming to optimize resource usage and minimize service delay. The paper considers metrics such as efficient resource utilization, low service delay, and scalability to address the challenges of deploying edge servers in ECC systems. The proposed policy demonstrates significant improvements over random deployment strategies through simulations, highlighting the importance of intelligent deployment decisions in enhancing service performance in ECC environments.

3.3.2 Experimentation-based

The authors of [35, 164] propose the integration of SDN and container-based virtualization with MEC architecture to manage MEH and ensure end-to-end mobility support for mobile users. The paper focuses on service migration between MEHs of different networks, emphasizing SDN's role in handling mobility challenges and ensuring QoS. The proposed architecture is validated through V2X simulations, demonstrating the benefits of centralized network intelligence and the modularity of SDN and containers for improved service continuity and QoE in scenarios like V2X mobility.

The authors of [165] integrate SDN with MEC to enable dynamic relocation of communication endpoints from the core to edge infrastructure in cellular networks. Real-world experiments on an NFV-based testbed evaluate session continuity and latency reduction during endpoint relocation.

The authors of [166] tackle the challenge of frequent user mobility in MEC by proposing a solution for seamless application migration between multiple MEC sites. The paper introduces a metaheuristic Tabu search algorithm for optimizing state transfer and the Flexible and Low-Latency State Transfer (FAST) framework, which directly forwards states based on SDN. Simulation and practical testbed results demonstrate the effectiveness of both approaches in reducing migration costs and ensuring efficient service deployment.

The authors of [167] focus on leveraging MEC and NFV in the context of 5G networks to enable flexible placement and migration of VNFs. The paper addresses the challenges of latency constraints and resource optimization by distributing computational and network resources closer to end users through a multi-tier cloud architecture. The study proposes an NFV-enabled testbed implementation to validate the architecture's effectiveness in managing cloud and edge resources efficiently, particularly in meeting the demanding requirements of 5G scenarios.

The authors of [168] focus on managing Cooperative Adaptive Cruise Control (CACC) for connected cars through MEC, addressing latency and network variability challenges. The paper introduces a context-aware Q-Learning migration scheme for the platoon controller, enabling dynamic adaptation to changing network conditions. Additionally, it proposes an asynchronous shared learning algorithm for rapid policy convergence. It evaluates the performance through simulations, showcasing better adherence to speed and spacing presets compared to existing migration schemes.

In Subsection 3.3.1, the algorithm-based evaluations primarily assess algorithmic performance and feasibility within controlled environments. However, these algorithms may not fully encapsulate the complexities and uncertainties present in real-world scenarios. Conversely, the experimental evaluations discussed in Subsection 3.3.2 provide valuable insights into real-world performance and practical challenges. However, the efficiency of algorithms is often constrained by testbed availability and may not encompass all possible scenarios. Both algorithmic and experimental approaches need to be considered together. A combined approach is necessary for leveraging algorithmic information, and experimentation is essential to comprehensively evaluate resource allocation strategies. This bridges the gap between theoretical analysis and real-world deployment, ensuring optimal system performance in 5G-MEC environments.

Chapter 4

Research Contributions

This chapter is divided into two sections. The first section initially introduces the general motivation of the research activity and presents the Research Questions (RQs) and the related Research Objectives (ROs). Finally, the research methodologies that are used to address the RQs are introduced. The second section highlights the primary contributions of the papers, elucidating the connections between the papers, RQs, and the adopted research methodologies. Additionally, a more detailed summary of the paper's research contributions is provided.

4.1 Research Design

4.1.1 Motivation

The development of 5G-MEC testbeds is indispensable for advancing V2X communications. 5G-MEC testbeds are particularly important because a 5G-MEC system has critical requirements such as ultra-low latency, high bandwidth, scalability, edge intelligence, and network slicing. A testbed can show the feasibility of such systems and highlight the potential benefits of specific use cases, such as automotive. By processing data at the network edge, 5G-MEC testbeds can be used to show significant latency reduction, ensuring real-time responsiveness, which is crucial for applications like autonomous driving and emergency collision avoidance. Additionally, these testbeds evaluate robust bandwidth support, scalable architectures, and edge intelligence capabilities, empowering V2X applications with advanced functionalities such as predictive analytics and dynamic resource allocation.

The development of 5G-MEC testbeds is complex since these testbeds integrate network technologies with computing technologies. Moreover, these testbeds are heavily based on virtualization and sofwarization and require advanced orchestration and management. They therefore require a broad knowledge, but they usually focus on a specific use case. Within the development of a 5G-MEC testbed, the PhD research activity mainly focuses on the following aspects:

- Integrating open-source solutions and software packages for 5G and MEC,
- Ensuring robust support for 5G-MEC app during user mobility,
- Delving into 5G-MEC app support in V2X use cases.

Open-source solutions, such as OpenStack, K8s, or Apache, can offer opportunities to streamline the deployment and management of MEC infrastructure. These platforms offer robust orchestration, virtualization, and containerization capabilities, enabling efficient resource allocation and scalability for MEC applications. By using ETSI-MEC OpenAPIs, the aim is to seamlessly embed MEC capabilities into the fabric of the 5G network, enabling ultra-low latency, high bandwidth, and localized data processing. This integration not only enhances the performance of latency-sensitive applications, like real-time gaming, High Definition (HD) video streaming, IoT applications, and industrial automation but also lays the groundwork for emerging technologies such as autonomous vehicles and smart cities. Ensuring robust support for MEC applications/services during user mobility within the dynamic 5G environment is essential for maintaining uninterrupted connectivity and service continuity for the end user. This entails developing efficient handover mechanisms, dynamic resource allocation strategies, and adaptive network management protocols to seamlessly deal with user movements across different network cells and RATs. MEC may support V2X use cases, such as MEHs hosting V2X MEC services to offer computational resources and low-latency communication to end users. During vehicle mobility, continuous MEC application/service connectivity needs to be ensured, while dynamic resource allocation optimizes resource usage to meet varying communication demands. Meanwhile, application migration within MEC can enable flexible deployment and management of V2X services, adapting to changing conditions and maintaining service continuity for the end-user.

4.1.2 Research Questions and Research Objectives

This thesis delves into exploring various facets of 5G-MEC systems. The research goals come from the necessary requirements for setting up 5G-MEC testbeds, how the integration of 5G and MEC affects resource availability, challenges in maintaining uninterrupted service during migrations and improving MEH selection and traffic path computation. Additionally, each RQ is addressed by RO based on the research contributions of the thesis.

RQ1: *How can a 5G-MEC testbed for V2X applications be implemented?*

RO1: This RO focuses on investigating the current landscape of 5G-MEC testbeds for V2X applications. Firstly, establishing the hardware setup is crucial. Many 5G-MEC testbeds have been developed in recent years. These testbeds are focusing on different use cases and some of them are related to V2X applications. There are also testbeds of V2X applications that focus on 5G but do not use MEC solutions. These testbeds can be based on closed solutions provided by a partner company or can integrate open-source solutions at different levels. This variegated landscape of diverse technologies and approaches should be investigated to determine which is the most effective way to implement a 5G-MEC testbed nowadays. This investigation first includes an understanding of how to set up the necessary hardware infrastructure: the 5G network along with the MEC system, which includes servers equipped with high-performance processors, memory, and storage. Secondly, the software setup is essential. A 5G-MEC testbed requires the installation of various software components like the MEP software enabling edge computing capabilities, V2X application software tailored for communication between vehicles, roadside infrastructure, and other entities, and the necessary networking protocols to facilitate seamless communication. Integration of hardware and software is vital for a cohesive testbed environment. This entails analyzing existing testbeds and understanding software functionalities for an effective implementation.

RQ2: *How can Kubernetes be used in a 5G-MEC infrastructure to orchestrate the MEC application/service deployment?*

RO2: To orchestrate and manage the deployment of MEC applications/services within a 5G-MEC infrastructure, K8s is employed for its KPIs, facilitating containerization and enabling deployment automation,

thereby simplifying deployment, scaling, and updating processes beneficial for MEC applications/services. Despite its benefits, K8s fail to fully meet the dynamic requirements of MEC applications tailored to end-user demands. Hence, additional strategies have to be explored to supplement K8s functionalities and overcome its limitations. These strategies also include the migration mechanisms to move the MEC applications between MEHs in order to maintain the performance and dependability experienced by the end users. This comprehensive approach should ensure the efficient resource utilization of the 5G-MEC system, which is challenging in a dynamic environment such as V2X communications.

RQ3: *How can a multi-objective allocation of data and network resources be performed and evaluated in a 5G-MEC testbed?*

RO3: A 5G-MEC system includes both data resources, which are related to the computing and storage capabilities of the MEC, and network resources provided by 5G. These resources are used to provide heterogeneous services that 5G is able to provide thanks to network slicing. This implies that these resources have to be effectively allocated considering multiple performance objectives. These resource allocations include the MEC service migration, the handover between Point of Access (PoAs), and the path selection between MEH and the user. All these allocation approaches have to be tested in an actual 5G-MEC testbed in order to also evaluate the QoE of the end users.

RQ2 and RQ3 are closely related, with RQ2 focusing on the implementation of a 5G-MEC testbed, while RQ3 focuses on the evaluation of solutions for allocating data and network resources. Overall, RQ2 and RQ3 complement each other aiming to provide a cohesive framework for the orchestration of 5G-MEC systems for V2X applications, where the solutions developed for RQ3 are evaluated in the testbeds developed for RQ2, and the potential of the testbeds is shown by applying the solutions.

4.1.3 Research Methodology

To answer the RQs that have been presented, several heterogeneous research methodologies should be used. These methodologies span from the classical literature review to the development of a testbed and to the proposal and evaluation of solutions for resource allocation.

- *Literature Review*: Conducting a literature review involves an in-depth analysis of existing works and state-of-the-art frameworks pertinent to deploying 5G-MEC testbeds. This encompasses understanding the nuances of functionalities, structural aspects, hardware requirements, and the intricate data flow within the system. Additionally, the review delves into scrutinizing experimental designs previously conducted in the field, providing valuable insights for subsequent implementation efforts. Drawing from the literature, hypotheses are developed to explore potential improvements in MEC systems. These include expectations for enhanced system performance and user experience through optimized resource allocation and migration strategies. The anticipated research artifacts consist of a hybrid 5G-MEC testbed, advanced algorithms for resource management, and tools that facilitate seamless application migration and handover.

- *Hardware Selection and Configuration*: Careful selection of hardware components is crucial for the efficient deployment and operation of the testbed. This includes choosing appropriate servers, network equipment, and edge computing devices and configuring them to meet the specific requirements of the 5G-MEC environment. The hardware setup undergoes iterative refinement based on detailed performance evaluations. Adjustments are made in response to specific requirements of the solution. For instance, when setting up a 5G-MEC emulator scenario, hardware capacity is selected to meet the performance demands of the emulation. This involves choosing servers with adequate CPU, memory, and storage resources to handle the expected load and data throughput. The selection process includes evaluating server specifications such as processor type, number of cores, RAM size, and network interface capabilities to ensure they align with the needs of the 5G-MEC system being tested.

- *Software Selection, Integration, and Development*: Selecting, integrating, and customizing the necessary software components are a significant aspect of the realization of a testbed. The selection and integration of existing open-source solutions, like OpenStack, K8s, and Docker, require careful adaptation to support MEC functionalities. By leveraging these solutions and customizing them to meet MEC requirements, one can benefit from established infrastructure and deployment frameworks while ensuring flexibility and scalability. Once the selected software has been integrated, the software development

of new features that enable new advanced functionalities is the final target. Validation is performed through the deployment and testing of the integrated software in both simulated and experimental environments. This process involves evaluating the software's performance across a range of dynamic application scenarios and resource allocation strategies. Key aspects of the validation include assessing the software's ability to handle varying loads, manage resources efficiently, and adapt to real-time changes in the network environment.

- *Algorithm Development*: Developing algorithms for the resource allocation within 5G-MEC system is another important aspect. These algorithms can be based on different techniques, from heuristics to mathematical optimization and machine learning. These algorithms should be tailored to the addressed scenario and aim to optimize system performance and enhance overall user experience. The algorithms are evaluated by using various performance metrics, such as latency and resource utilization, in different scenarios. The effectiveness of these algorithms is compared against conventional methods to verify improvements in efficiency and user experience.
- *Testing and Performance Evaluation*: Evaluating the performance of a testbed or of the developed algorithm is essential for assessing the effectiveness and feasibility of the proposed solution in a 5G-MEC system. Selecting the relevant metrics and utilizing tools, such as Grafana, Prometheus, Wireshark, application statistics analysis, K8s performance monitoring, and resource consumption tracking, enables comprehensive data gathering and insights into system behavior, aiding in optimization and troubleshooting efforts. Throughout the testing phase, data collected from performance evaluations informs ongoing adjustments and refinements. This iterative process ensures that the system and algorithms are continuously improved based on real-time feedback and performance results.

4.2 Contributions of the Papers

This section presents the main contributions of the thesis. As already mentioned, the thesis includes five publications. A mapping of the included papers, the research methodologies, and the RQs are presented. Then, a summary of each paper, where the research contributions are highlighted, is introduced.

4.2.1 Mapping Papers to RQs and Research Methodologies

Table 4.1 gives a clear overview of the research methodology used in each paper. Some of the research methodologies presented in the previous section have been grouped together. Table 4.2 shows the contribution of not included paper in the thesis.

Figure 4.1 visually presents which RQs are addressed by each paper and how the papers are connected with each other. Paper 1 provides a comprehensive overview of 5G-MEC testbeds for V2X applications, forming the basis for subsequent research. Based on the work performed for Paper 1, Paper 2 uses an emulation platform to evaluate handover strategies, addressing RQ1 and RQ2 by emphasizing practical evaluation over theoretical models. Paper 3 builds on Paper 2's findings to explore MEC application migration, leveraging Kubernetes for orchestration and addressing dynamic resource management, thus linking RQ1 and RQ3. Paper 4 tackles multi-objective optimization for MEC host selection and traffic path computation, aligning with RQ2 and RQ3 by proposing an advanced algorithm for efficient resource allocation. Finally, Paper 5 integrates insights from previous papers to develop a hybrid testbed for MEC application migration, demonstrating practical benefits and connecting RQ2 and RQ3. This interconnected network of papers forms a comprehensive framework for V2X applications, showcasing progressive research development from testbed implementation to orchestration and resource allocation strategies.

Table 4.1: Mapping of included Papers with the research methodology in Section 4.1.3.

Paper	**Research Methodology**		
	Study of the state of the art	5G-MEC Testbed Deployment	Algorithmic Solution Evaluation
Paper 1: 5G-MEC Testbeds for V2X Applications	✓	-	-
Paper 2: Experimental Evaluation of Handover Strategies in 5G-MEC Scenario by using Advant-EDGE	-	✓	✓
Paper 3: MEC Application Migration by using AdvantEDGE	-	✓	
Paper 4: Joint multi-objective MEH selection and traffic path computation in 5G-MEC systems	-	✓	✓
Paper 5: MigraMEC: Hybrid Testbed for MEC App Migration	-	✓	

Table 4.2: Mapping of not included Paper with the research methodology in Section 4.1.3.

Paper	Research Methodology		
	Study of the state of the art	5G-MEC Testbed Deployment	Algorithmic Solution Evaluation
Paper 6: Evaluating the Performance of a Real-Time VRU Detection System Using Edge Devices	-	✓	

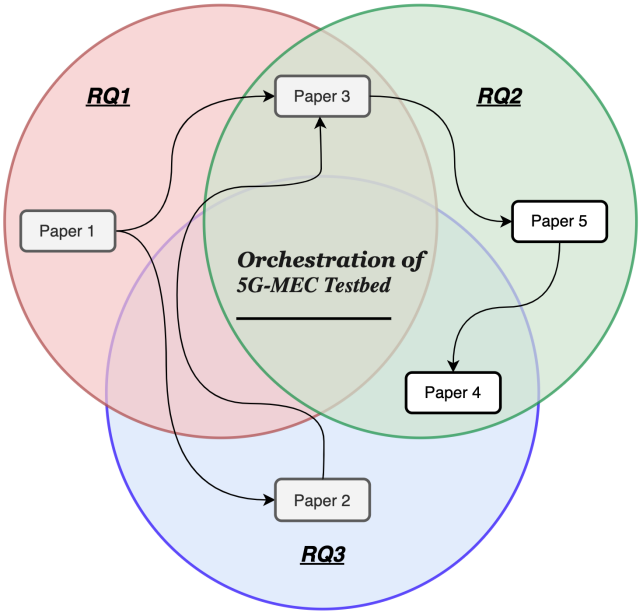


Figure 4.1: Overview of the RQs addressed by the included Papers and of the relationship between the Papers.

4.2.2 Paper 1

Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni "5G-MEC Testbeds for V2X Applications", MDPI Journal of Future Internet, 2023.

The primary focus of this paper is to offer a comprehensive overview of existing 5G-MEC testbeds with a particular focus on the context of V2X applications. Initially, the research delves into the state-of-the-art of 5G-MEC systems and proposes an overall system perspective based on discussions from standardization bodies and the requirements of MNOs.

Subsequently, it examines and categorizes existing 5G-MEC testbeds tailored for V2X applications, providing analysis through various use case scenarios. The study also evaluates the complexity of testbed usage by exploring factors such as testbed resources, replication methods, available open-source tools, and their associated limitations. Finally, it summarizes the available tools for the functionalities of 5G-MEC testbeds and discusses compatibility issues, including integration challenges and the development of open APIs.

4.2.3 Paper 2

Rosario G. Garroppo, Marco Volpi, Gianfranco Nencioni, Prachi V. Wadatkar "Experimental Evaluation of Handover Strategies in 5G-MEC Scenario by using AdvantEDGE", Mediterranean Conference on Communications and Networking (MeditCom), 2022.

Paper 1 presents an open-source solution for constructing the 5G-MEC framework and facilitating testing. Conversely, Paper 2 delves into an emulation platform tool called AdvantEDGE recommended by ETSI for testing various network models and MEC APIs. Utilizing AdvantEDGE, Paper 2 conducts experimental analyses focusing on different strategies for the handover between PoAs triggered by user mobility, adopting a multi-objective approach. Specifically, the testbed deploys a video streaming server as an MEC application on an MEH while the end users act as clients. The analysis captures the performance of multi-objective handover strategies through measured network data statistics from the video streaming application, and the QoE observed by end-users. This highlights the drawbacks of simplified theoretical or simulation models in accurately predicting the performance of tested strategies, algorithms, protocols, and applications, which consequently impacts the QoE.

4.2.4 Paper 3

Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni "MEC Application Migration by using AdvantEDGE", 17^{th} EAI International Conference on Tools for Design, Implementation and Verification of Emerging Information Technologies (TRIDENTCOM), 2022.

Paper 3 explores the feasibility of deploying multiple MEH instances within an emulated environment. Building on the strategies discussed in Paper 2, which enhanced QoE by rerouting traffic to the MEH, Paper 3

introduces an additional MEH into the scenario. The primary challenge analyzed is the dynamic switching from one MEH to another depending on user location, requiring seamless migration and orchestration of MEC applications and services while maintaining connectivity with the end users. K8s serves as the VIM to manage multiple MEH instances. A virtual MEO utilizes MEC API location information to conduct actual migration based on user location. The default K8s solution does not inherently support migration activities of this nature. Therefore, we propose integrating a prototype solution to facilitate such migrations. This analysis showcases the adept handling of MEC applications in dynamic environments.

4.2.5 Paper 4

Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni, Marco Volpi "Joint multi-objective MEH selection and traffic path computation in 5G-MEC systems", Elsevier Journal of Computer Networks, 2024.

Paper 4 addresses the MEH selection and the traffic path computation in 5G-MEC systems by defining a multi-objective optimization problem. The heterogeneity of 5G-MEC applications necessitates novel control algorithms to consider various criteria concurrently, such as latency minimization, packet loss reduction, security enhancement, and energy consumption minimization. Traditional single-criterion optimization methods are insufficient due to the complexity of these diverse requirements. Instead, the paper proposes a multi-objective optimization approach, generating a Pareto front of optimal solutions dynamically selected based on application needs. The MEO analyzes this Pareto front to determine the most suitable solution for application requirements, facilitating MEH placement and traffic steering. Despite being NP-hard, recent works offer efficient algorithms for multi-criteria optimization in practical scenarios. The paper's key contributions include defining a graph to jointly address MEH selection and traffic path computation problems, introducing the MDA to compute the Pareto front, demonstrating its effectiveness across various network and application scenarios, and integrating MDA into a hybrid testbed for evaluation in simulated, emulated, and experimental environments. Additionally, the paper develops a controller to interface with MDA, enabling seamless integration with the 5G-MEC system at both network and application layers.

Paper 4 assumes constant link attributes in the MDA algorithm, which may not fully capture the dynamic nature of wireless networks where link attributes between UE and PoAs can vary due to cell load, channel

conditions, and interference. To address this, the paper suggests considering approaches such as treating each PoA as a potential source node and dynamically recalculating non-dominated paths to adapt to time-varying attributes. Additionally, incorporating MEH performance information through additional arcs to model data exchange with the network layer is explored. This approach, combined with real-time monitoring tools and load balancing strategies, ensures that the MDA algorithm remains effective in dynamic environments.

4.2.6 Paper 5

Prachi V. Wadatkar, Rosario G. Garroppo, Gianfranco Nencioni "MigraMEC: Hybrid Testbed for MEC App Migration", 29th Annual International Conference on Mobile Computing and Networking (MobiCom), 2023.

Paper 5 presents MigraMEC, a hybrid testbed architecture combining network simulation, user mobility emulation, and MEC framework integration via AdvantEDGE. It employs an extended version of K8s to implement two physical MEHs. The MigraMEC controller orchestrates interactions between emulated and experimental environments for efficient MEC application migration. Leveraging network information, it selects appropriate MEHs and PoAs for users. Using a video streaming service, the demonstration underscores the importance of multiple MEHs and streamlined migration for optimal user experience.

Chapter 5

Conclusion

The primary objective of this research was to leverage the capabilities of emerging technologies like 5G and MEC to establish an efficient, automated system for managing and orchestrating services and resources for V2X applications. The aim was to achieve low latency, higher throughput, better QoE, and seamless orchestration of a 5G-MEC system with multiple heterogeneous targets.

In this chapter, the main discoveries and conclusions of this thesis are summarized. Additionally, the future directions of this research are outlined.

5.1 Main Findings

The primary discoveries of each included paper, which address the RQs and ROs presented in the previous chapters, can be summarized as follows.

- **Paper 1** discusses the importance of 5G-MEC solutions in delivering low-latency applications and enhancing user experience. The paper identifies gaps between ETSI-defined deployment models and practical solutions, focusing on integration challenges, compatibility concerns, and interface development among MEC elements. The paper emphasizes the need for open-source tools/platforms integration, platforms for 5G-MEC V2X simulation, and compatibility among MEC building block tools.

- **Paper 2** performs an experimental analysis that highlights the effectiveness of the AdvantEDGE emulator in maintaining QoE under varying network conditions. The paper introduces a multi-objective strategy for the handover between PoAs with a minimum throughput guarantee, showcasing its benefits in UE movement scenarios and network degradation situations.
- **Paper 3** emphasizes the essential integration of K8s within the MEC framework, highlighting its pivotal role in managing MEC applications. The paper investigates migration strategies and integrates prototype solutions for addressing the challenge of user mobility and satisfying the consequent need for application migration to uphold QoE standards.
- **Paper 4** proposes an MDA-based solution for selecting MEHs and paths between UE and MEHs, showing MDA's effectiveness in optimizing network performance and user experience during application migration through hybrid testbed evaluation. This approach comprehensively evaluates potential MEH-path combinations based on various criteria, such as network-layer and application-layer metrics. Evaluation scenarios with two and one MEHs show the MDA controller's ability to migrate MEC applications with minimal impact on network performance and user experience, highlighting its crucial role in optimizing MEC environments. The evaluation considers essential network performance parameters such as latency and throughput, and the user experience is assessed through Mean Opinion Score (MOS).
- **Paper 5** described a network scenario involving different PoAs and UE configurations. It demonstrates the impact of MEH availability on video quality and network performance, highlighting the benefits of having multiple MEHs for optimized QoE during application migration.

In summary, these papers collectively address various aspects of 5G-MEC solutions, including deployment challenges, QoE maintenance, application migration strategies, network performance optimization, and the advantages of edge computing in enhancing user experiences. Each paper contributes to a broader understanding of the complexities and opportunities in the field of MEC.

In conclusion, this thesis enhances the understanding of integrating 5G and MEC systems for V2X applications by tackling both theoretical and

practical challenges. It highlights the necessity of developing a hybrid testbed as a practical approach to navigate the complexities of 5G-MEC orchestration and resource management. By bench marking existing 5G MEC solutions, the research addresses the need to manage significant heterogeneity in resources, services, vendors, and the high dynamism of network traffic, alongside the elevated mobility of users in vehicular communication.

To address these challenges, the study offers a comprehensive feature-based analysis of existing testbeds, categorizing them into three distinct aspects: 5G-MEC, 5G-V2X, and 5G-MEC-V2X. This analysis demonstrates that a hybrid solution is the most feasible in terms of cost-effectiveness and resource utilization. Additionally, the thesis provides detailed information on software packages and SDKs available for each component within the 5G-MEC-V2X testbed. It also discusses open challenges, particularly the development of APIs, which is a crucial and widely debated aspect.

In the realm of MEC application orchestration, the thesis proposes the use of K8s, a leading solution for VIMs, to containerize MEC services and applications. This approach simplifies and controls deployment. Furthermore, the research enhances the existing K8s solution by integrating a prototype to address migration and handover requirements, which are critical for user mobility, especially in V2X applications.

Finally, the proposed multi-objective resource allocation approach, utilizing the MDA algorithm, represents a significant advancement in optimizing network performance and user experience. By evaluating these strategies in a hybrid testbed, this work not only validates theoretical models but also provides practical insights for future developments in 5G-MEC systems.

5.2 Future Direction

The future direction for enhancing the thesis's contribution involves integrating computational resource metrics from the MEH and considering resource availability within the optimization algorithm in dynamic environments. These computational resources can be sourced from the K8s manager responsible for managing the MEH and deploying MEC applications based on user demand. Factoring in resource availability for deploying heavy computational applications can significantly improve efficiency.

Furthermore, the development of the MigraMEC controller holds promise for maturing its capabilities in managing these responsibilities. The optimization algorithm can make more informed decisions by leveraging real-time data on computational resources and availability, leading to better

resource allocation and overall system performance. Additionally, exploring mechanisms to dynamically scale resources based on fluctuating demand and workload requirements can enhance the adaptability and scalability of the system in dynamic environments. Strategic integration of optimization algorithms based on computational resource consideration can improve resource allocation within the 5G-MEC framework. These algorithms optimize resource utilization and enhance overall system efficiency by analyzing real-time data on network traffic, application requirements, and user behavior. The framework may dynamically allocate resources by utilizing historical data and real-time conditions, addressing scalability and agility challenges.

The continuous evolution and integration of SDN and NFV in the MEC architecture can significantly benefit resource allocation optimization [169]. SDN allows for centralized management and orchestration of network resources, facilitating dynamic adjustments based on application demands and network conditions. At the same time, NFV enables flexible deployment of NFs, ensuring efficient resource utilization. Enhanced SDN capabilities further improve resource allocation and network management by utilizing Artificial Intelligence (AI)-driven SDN controllers for centralized orchestration, optimizing network resources across distributed edge nodes. Furthermore, SDN-based controllers and NFV orchestrators will facilitate dynamic service orchestration in the 5G-MEC environment, allowing for the automated deployment, scaling, and management of MEC applications. This automation streamlines operations reduces deployment times and improves agility in adapting to changing application demands and network conditions.

Looking forward, the creation of fully autonomous networks incorporating advanced management and orchestration features such as self-optimization, experimental configuration, and connectivity restoration will be essential [170]. Current automation solutions are external, but the evolution towards 6G aims for seamless global connectivity through space-air-ground integrated networks (SAGIN). As technology advances, the integration of non-terrestrial networks with onboard processing will boost both capacity and computational capabilities. Managing these complex networks will require sophisticated resource management strategies to handle dynamic topologies and diverse service demands [171].

The enhancement of control functions with computing, caching, and communication (4C) will be critical for network management. Context-aware

communication strategies leveraging computation and caching are necessary. Advances in semantic communications will reduce data transmission and improve efficiency. Furthermore, AI-driven solutions for optimizing resource allocation and addressing emerging applications will be crucial for advancing next-generation networks.

Part II

Included Papers

Paper 1:
5G-MEC Testbeds for V2X Applications

5G-MEC Testbeds for V2X Applications

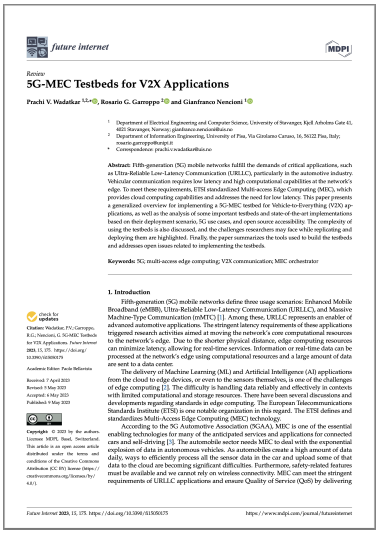
Prachi V. Wadatkar[1,2], Rosario G. Garroppo[2], Gianfranco Nencioni[1]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger

[2] Department of Information Engineering, University of Pisa

Published by the MDPI Journal of Future Internet, 2023

https://www.mdpi.com/1999-5903/15/5/175



future internet

MDPI

Review

5G-MEC Testbeds for V2X Applications

Prachi V. Wadatkar [1,*], Rosario G. Garroppo [2] and Gianfranco Nencioni [1]

1 Department of Electrical Engineering and Computer Science, University of Stavanger, Kjell Arholms Gate 41, 4021 Stavanger, Norway; gianfranco.nencioni@uis.no
2 Department of Information Engineering, University of Pisa, Via Girolamo Caruso, 16, 56122 Pisa, Italy; rosario.garroppo@unipi.it
* Correspondence: prachi.v.wadatkar@uis.no

Abstract: Fifth-generation (5G) mobile networks fulfill the demands of critical applications, such as Ultra-Reliable Low-Latency Communication (URLLC), particularly in the automotive industry. Vehicular communication requires low latency and high computational capabilities at the network's edge. To meet these requirements, ETSI standardized Multi-access Edge Computing (MEC), which provides cloud computing capabilities and addresses the need for low latency. This paper presents a generalized overview for implementing a 5G-MEC testbed for Vehicle-to-Everything (V2X) applications, as well as the analysis of some important testbeds and state-of-the-art implementations based on their deployment scenario, 5G use cases, and open source accessibility. The complexity of using the testbeds is also discussed, and the challenges researchers may face while replicating and deploying them are highlighted. Finally, the paper summarizes the tools used to build the testbeds and addresses open issues related to implementing the testbeds.

Keywords: 5G; multi-access edge computing; V2X communication; MEC orchestrator

1. Introduction

Fifth-generation (5G) mobile networks define three usage scenarios: Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communication (URLLC), and Massive Machine-Type Communication (mMTC) [1]. Among these, URLLC represents an enabler of advanced automotive applications. The stringent latency requirements of these applications triggered research activities aimed at moving the network's core computational resources to the network's edge. Due to the shorter physical distance, edge computing resources can minimize latency, allowing for real-time services. Information or real-time data can be processed at the network's edge using computational resources and a large amount of data are sent to a data center.

The delivery of Machine Learning (ML) and Artificial Intelligence (AI) applications from the cloud to edge devices, or even to the sensors themselves, is one of the challenges of edge computing [2]. The difficulty is handling data reliably and effectively in contexts with limited computational and storage resources. There have been several discussions and developments regarding standards in edge computing. The European Telecommunications Standards Institute (ETSI) is one notable organization in this regard. The ETSI defines and standardizes Multi-Access Edge Computing (MEC) technology.

According to the 5G Automotive Association (5GAA), MEC is one of the essential enabling technologies for many of the anticipated services and applications for connected cars and self-driving [3]. The automobile sector needs MEC to deal with the exponential explosion of data in autonomous vehicles. As automobiles create a high amount of data daily, ways to efficiently process all the sensor data in the car and upload some of that data to the cloud are becoming significant difficulties. Furthermore, safety-related features must be available and we cannot rely on wireless connectivity. MEC can meet the stringent requirements of URLLC applications and ensure Quality of Service (QoS) by delivering

Future Internet 2023, 15, 175. https://doi.org/10.3390/fi15050175 https://www.mdpi.com/journal/futureinternet

Check for updates

Citation: Wadatkar, P.V.; Garroppo, R.G.; Nencioni, G. 5G-MEC Testbeds for V2X Applications. Future Internet 2023, 15, 175. https://doi.org/10.3390/fi15050175

Academic Editor: Paolo Bellavista

Received: 7 April 2023
Revised: 5 May 2023
Accepted: 6 May 2023
Published: 9 May 2023

Abstract:
The Fifth-Generation (5G) mobile networks fulfill the demands of critical applications, such as Ultra-Reliable Low-Latency Communication (URLLC), particularly in the automotive industry. Vehicular communication requires low latency and high computational capabilities at the network's edge. To meet these requirements, ETSI standardized Multi-access Edge Computing (MEC), which provides cloud computing capabilities and addresses the need for low latency. This paper presents a generalized overview for implementing a 5G-MEC testbed for Vehicle-to-Everything (V2X) applications, as well as the analysis of some important testbeds and state-of-the-art implementations based on their deployment scenario, 5G use cases, and open-source accessibility. The complexity of using the testbeds is also discussed, and the challenges researchers may face while replicating and deploying them are highlighted. Finally, the paper summarizes the tools used to build the testbeds and addresses open issues related to implementing the testbeds.

6.1 Introduction

The Fifth-Generation (5G) mobile networks defines three usage scenarios: enhanced Mobile BroadBand (eMBB), Ultra-Reliable Low-Latency Communication (URLLC), and massive Machine-Type Communication (mMTC) [172]. Among these, URLLC represents an enabler of advanced automotive applications. The stringent latency requirements of these applications triggered the research activities aimed at moving the network's core computational resources to the network's edge. Due to the shorter physical distance, edge computational resources will minimize latency, allowing for real-time services. Information or real-time data will be processed at the network's edge using computational resources, and a large amount of data will be sent to the data center.

Delivering Machine Learning (ML) and Artificial Intelligence (AI) from the cloud to edge devices, or even to the sensors themselves, is the challenge in edge computing [173]. The difficulty is handling data reliably and effectively in contexts with limited computational and storage resources. There have been several discussions and developments to build a standard for edge computing; European Telecommunications Standards Institute (ETSI) is one. ETSI defines and standardizes Multi-access Edge Computing (MEC) technology.

As per the 5G Automotive Association (5GAA), MEC is one of the essential enabling technologies for many of the anticipated services and applications for connected cars and self-driving [174]. The automobile sector needs MEC to deal with the exponential explosion of data in autonomous vehicles. As automobiles create a high amount of data daily, efficiently processing all that sensor data in the car and uploading parts of that data to the cloud is becoming a significant difficulty. Furthermore, safety-related features must be available, and we cannot rely on wireless connectivity to work. MEC can meet the stringent requirements of URLLC applications and maintain the Quality-of-Service (QoS) by delivering the MEC services to the application developers and content suppliers. MEC allows operators to access the cloud infrastructure. Regardless of the underlying Radio Access Network (RAN) (e.g., 3rd Generation Partnership Project (3GPP) [84]), MEC access agnostic nature makes the deployment effortless.

Vehicle-to-Everything (V2X) applications benefit from both 5G and MEC [175]. Vehicles may connect with the Road Side Unit (RSU) infrastructure and other vehicles around them using this technology. The vehicle communicates with compatible equipment via a short-range wireless signal

immune to interference and lousy weather. V2X technologies are primarily employed to improve safety and avoid collisions. Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N), Vehicle-to-Pedestrian (V2P), Vehicle-to-Device (V2D), Vehicle-to-Grid (V2G), and Vehicle-to-Vehicle (V2V) are the primary forms of V2X technology [176]. V2I technology gives real-time infrastructure data to drivers, including road conditions, traffic congestion, parking availability, and accidents. V2V communication allows cars to transmit messages to each other via a wireless network with data about what they are doing. Speed, position, braking, direction of motion, and loss of stability are all included in the data. Dedicated Short-Range Communications (DSRC) technology is enabled, which is equivalent to Wi-Fi [177]. Almost all road safety services and most traffic management and efficiency services need the periodic transmission of V2V and V2I messages with extremely low latency and high dependability [178].

The performance of the V2X applications depends on many parameters at different layers of the protocol stack: from the physical problems related to the 5G radio link conditions to the networking issues related to the QoS guarantee of the exchanged data to the application problems related to the storage and computation resources available for completing the application tasks. To account for the large set of parameters, testbeds have been deployed to experiment with the V2X application without the approximation added by theoretical or simulation models.

6.1.1 Contribution of the Paper

This paper is inspired and motivated by the development and deployment strategies conducted for 5G and MEC systems to support V2X applications. The survey [179] demonstrates MEC integration with new technologies, including 5G and beyond; in addition to that, it presents the open-source activities, testbeds, and implementations. MEC deployment for vehicular network applications and services is provided in [180]. 5G use case scenarios with the corresponding traffic models per the Standard Development Organizations (SDOs) and stakeholder requirements are presented in [6]. In [181], the authors discuss the MEC deployment of resources, its migration abilities, and use cases for industrial verticals, whereas in [182] the authors present capabilities of MEC orchestration, deployment, and open-source tools. In [93], the authors exhibit 5G testbeds for network slicing applications.

In concisely, the contribution of the paper is described as follows:

(i) To develop a 5G-MEC testbed for V2X applications experimentation, we investigate the literature on the 5G-MEC system and study its architecture. Based upon the generalized idea proposed by the standardized body and commercial vendor, such as Mobile Network Operators (MNOs) requirements, we suggest an overall view of the system, whereas the hardware and software components are considered. The software modules such as MEC platform (MEP), orchestration, and network flow are placed as per the industry standard and their responsibilities.

(ii) The paper presents the state of the art of the existing testbeds, shows how the 5G-MEC system is implemented, and emphasizes its use case scenarios by providing a short description. In the case of V2X application testbeds, we present a summary of the Vehicle User Equipment (VUE) and Road Side Unit (RSU) types of equipment.

(iii) The core part of the paper is studying the usage complexity of the surveyed testbeds and whether some of the testbeds are open-source and accessible. We investigate the difficulties of using the testbed resources, replicating the testbeds, and its limitations.

(iv) The paper summarizes the available tools used in the surveyed testbeds. Along with this, we present other open-source tools to deploy functionalities of the testbed. Since most functionalities can be deployed by using open-source tools, we discuss their compatibility issues with each other and how to integrate them by following the open-source community support.

6.1.2 Alignment of the Paper

The paper is organized as follows. Section 6.2 addresses the background knowledge of 5G aspects and features, and fundamentals of MEC, essential 5G use cases such as URLLC classified V2X application and a global view of the 5G-MEC testbed for vehicular communication. Section 6.3 involves the existing testbeds and implementations, where the surveyed testbeds are evaluated and segmented into 5G-MEC, 5G-V2X, and 5G-MEC for V2X testbeds. Section 6.4 describes the usage complexity of the surveyed testbeds. Section 6.5 outlines the summary of tools used to build the testbed. Finally, Section 6.6 concludes the paper and addresses unresolved issues. Figure 6.1 depicts the map and layout of this paper. We present the most used acronyms and their definitions at the end of the paper.

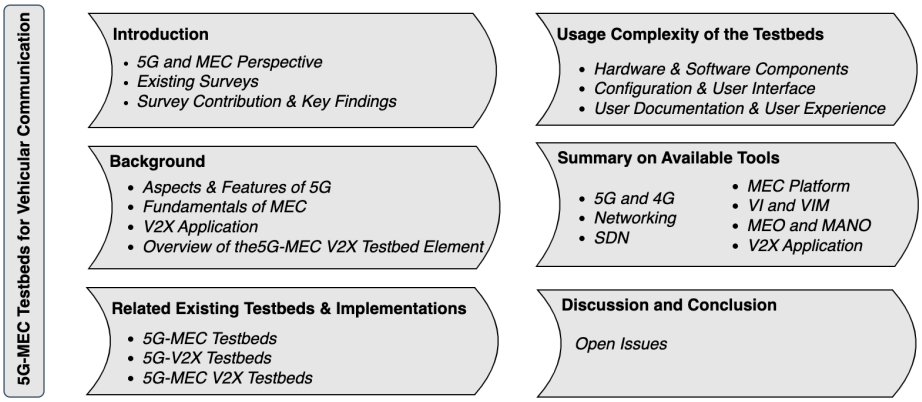


Figure 6.1: Map and layout of the paper.

6.2 Background

A MEC system creates the back-end support for 5G to achieve the critical latency application such as URLLC in which latency can require a maximum of $1ms$. The 5G-MEC paradigm can deliver the applications' latency and other critical requirements. The MEC system helps to analyze the real-time scenario in the case of the V2X application and can perform the required computation duties. The MEC system provides computation capabilities using virtualization and softwarization technologies such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV).

This segment provides insights into 5G-MEC and V2X applications in detail. The section is categorized into four parts discussing 5G features and capabilities, fundamentals of MEC, V2X application, and the general elements of the 5G-MEC V2X testbed. MEC fundamentals include integrating MEC deployment into 5G and softwarization technologies such as SDN and NFV with MEC. The 5G-MEC V2X testbed's general elements are divided into two parts: hardware and software components.

6.2.1 Aspects and Features of 5G

5G is growing as the new reference architecture in the telecommunication industry. The 5G network introduces modular Network Functions (NFs) for the control plane and user plane in both Access Network (AN) and Core

Network (CN) [183]. NFs provide specific network capabilities according to the application requirements. 5G will support different vertical sectors and industries having machine and human-type communication. 5G is expected to be native support of MEC to achieve low latency and more network efficiency.

The main usage scenarios of 5G are divided into three categories [184]: eMBB, which provides a data rate higher than 4G complemented by moderate latency improvements. eMBB supports the development of use cases such as 360-degree streaming video, Augmented Reality/Virtual Reality (AR/VR) media, and UltraHD applications. mMTC provides connectivity to an enormous number of devices. 5G will replace the previous technologies such as LTE-M, Low-Power Wide Area Network (LPWAN) technologies, and NarrowBand Internet of Things (NB-IoT). URLLC requires the 5G core deployment to reduce end-to-end (E2E) latency. URLLC also accommodates emerging services and applications with high-reliability requirements. URLLC enables the support of applications such as remote control and autonomous driving.

ITU-R M.2410 [4] presents the 5G requirements for the eMBB, mMTC, and URLLC applications. The user plane latency required by eMBB and URLLC is 4ms and 1ms, respectively. In addition, eMBB requires a 20Gbps downlink data rate, and for mMTC, a connection density of 10^6 devices per km^2.

The core of the 5G system is the service-based architecture (SBA), which defines the interaction between NFs responsible for the control plane and data plane procedures. Figure 6.2 depicts the non-roaming 5G system reference architecture. The bottom layer of the architectural elements is involved in transporting the user data, whereas the upper layer includes essential NFs within the control plane. The NFs within the 5G reference architecture are Access and Mobility Management Function (AMF), Session Management Function (SMF), Network Slice Selection Function (NSSF), Network Repository Function (NRF), Unified Data Management (UDM), Policy Control Function (PCF), Network Exposure Function (NEF), Authentication Server Function (AUSF) and User Plane Function (UPF). 3GPP TS 23.501 [3] entitles the functionalities and responsibilities in detail. In the 5G system, the separation of user plane functions and control plane functions leads to flexible deployment and scalability. The 5G system model is defined to enable technologies such as SDN and NFV and to provide data connectivity.

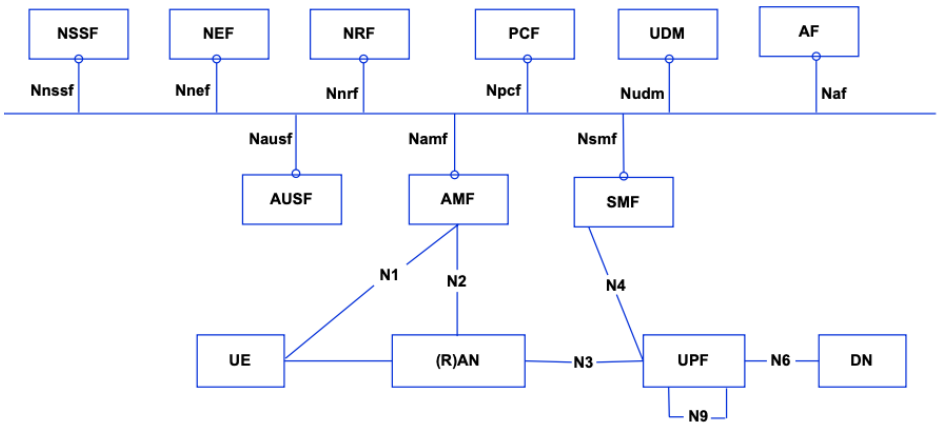


Figure 6.2: 5G system reference architecture.

6.2.2 Fundamentals of ETSI-MEC

URLLC and eMBB applications require computational capabilities, which becomes an inherent concern due to the end-users limited storage and computational abilities. Due to these minimum requirements, the 5G network will face unusual traffic volume. At the same time, MEC is an emerging concept in the 5G network that has the capacity to process a lot of data inside the RAN. MEC provides support to meet the requirements of the applications where real-time processing is needed. Fog computing and cloudlet are alternative architectures that, like MEC, include cloud capabilities to the edge of the networks. The core idea of MEC is to deploy the cloud computing capabilities within the RAN, close to the end-user [185]. ETSI is a standardized organization for the MEC and provides a reference architecture for deploying the functionalities. ETSI divides the MEC system into different MEC building blocks and discusses the contribution of each building block.

MEC Architecture

The MEC architecture gives an extensive high-level review of the MEC system, enabling the MEC applications to operate uninterrupted in the multi-access network. The MEC system reference architecture comprises a MEC management entity and one or more MEC hosts (MEHs) [17]. Figure 6.3 represents the MEC system reference architecture defined by

the ETSI. In the following, we present the different architectural elements (in this paper referred to as MEC building blocks)

MEC Host

MEH is an element that contains the MEC Platform (MEP) and a Virtualization Infrastructure (VI), which is a set of computing and storage resources, and the network for the MEC applications (MEC apps). The data plane in the VI performs the traffic rules received by the MEP. The MEP routes the traffic among other services, applications, and access networks. Usually, these functions are run on Virtual Machines (VMs), isolation supported by virtualization, and provide specific network services.

MEC Platform

The MEP is an integral part that delivers a set of tasks necessary for MEC apps to run on a specific VI, configures the DNS proxy and server, and provides an environment for both providing and leveraging MEC services. It is up to the MEP to grant access to storage and time-of-day data.

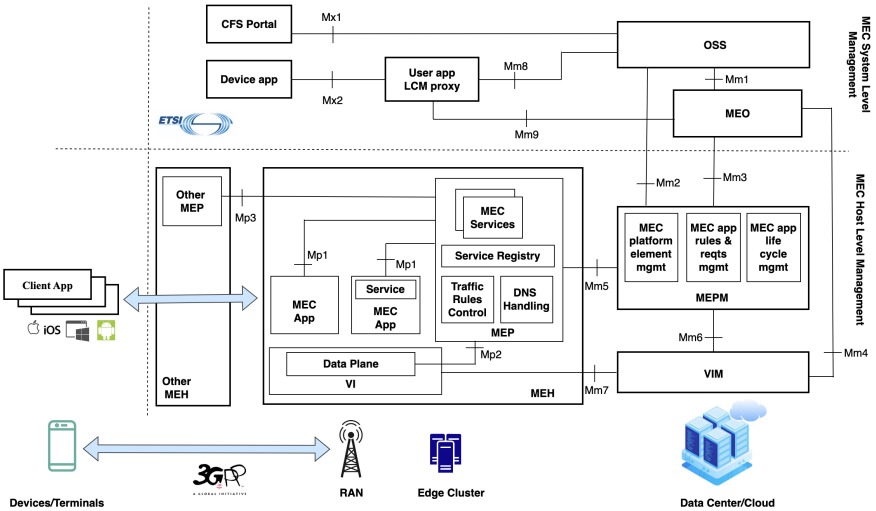


Figure 6.3: ETSI MEC reference architecture.

MEC Management

The MEC management consists of two components, the MEC system-level management and the MEC host-level management. MEC system-level management interacts with the MEC host-level management and the access network. The MEC system-level management consists of the following elements:

- *MEC Orchestrator (MEO)*, which is the brain of the MEC since it contains the complete global overview of the MEHs, services, resources, and topology. It determines the most appropriate MEH based on the user requirements. MEO can perform the system-level tasks with the Operating Support System (OSS) and the user application life-cycle management proxy. It can trigger application termination, relocation, and instantiation as needed.

- *OSS*, which receives a request via the Customer Facing Service (CFS) portal, offers the decision on these requests and the capacity to enable or terminate applications. For further processing, an MEO handles the requests. When supported, OSS receives a request to relocate applications between the MEC system and the cloud.

- *User Application Life-cycle Management Proxy*, a MEC application used for the life cycle management of UE applications. It permits device applications to trigger instantiation, onboarding, and application termination for users when supported. It is a function that can be accessed from the mobile network and can inform the state of the user application to the device application. For further processing of the requests, it interacts with the MEO and the OSS.

The MEC host-level management consists of the following elements:

- *MEC Platform Manager (MEPM)*, which is responsible for the set of applications such as providing the critical information to the MEO concerning MEP, giving the aspect of life cycle management of the MEC apps, configuring the DNS, managing the rules, and authorizing services.

- *Virtualization Infrastructure Manager (VIM)* deals with allocating and managing virtualized resources of the VI. It provides the virtual environment to host a VM and rapid provisioning of applications when supported. It gathers and reports the state of the performance and fault information in the virtualized resources.

5G and MEC Integration

MEC system interacts with the 5G SBA as shown in Figure 6.4. MEP is treated as an application function (AF) to access the 5G core network

[186]. UPF configuration and MEC deployment scenario depend upon the network and MEC provider. UPF integration into the MEC system is the primary component, which can be scattered and customizable data plane from the perspective of the MEC system. The traffic configuration of the data plane is done on the NEF-PCF-SMF route. MEH elements can be deployed in the data network of the 5G system, whereas NEF is deployed with MEO [187].

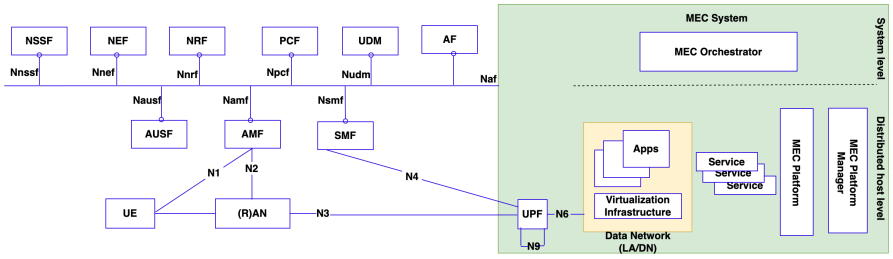


Figure 6.4: MEC deployment in 5G reference architecture.

Softwarization Technologies

Softwarization technologies are rapidly growing as the network paradigms for network and service providers. SDN and NFV are the key enabling technologies in the network paradigm and can support the 5G-MEC system.

SDN aims to provide network management, programmability, and flexibility of networks. By separating the control plane from the data plane, SDN allows network access seamless. The decoupling is performed via logically centralized network intelligence (SDN controller), while the data plane consists of network devices forwarding the packets. Google and Facebook have already introduced SDN in their respective data centers [32].

NFV allows network operators to provide network services by using the virtual infrastructure. NFV allows moving from deploying hardware-based proprietary network functions to deploying software network functions on off-the-shelf hardware. NFV provides the cost efficiency and the timely benefit of cloud computing. [188] presents the NFV reference architecture in which Virtual Network Functions (VNFs), NFV Infrastructure (NFVI), and Management and Network Orchestration (MANO) are the key elements.

MEC in NFV Environment

In the future, MNOs will deliver the network as a service (NaaS) by using NFV. MNOs introduce the VI to integrate the network elements, MEC apps, and components on the infrastructure [189]. The traffic is routed by the data plane between the applications, various services, access networks, and MEP in the VI. The data plane configuration is done via Mp2 reference [39]. The MEC and NFV architecture's fundamental component is VI. The MEPM shown in the MEC system reference architecture is transformed into a "Multi-access Edge Platform Manager-NFV" (MEPM-V) that transfers the Life-Cycle Management (LCM) section to one or more VNF Managers (VNFMs), which is smart of the NFV MANO. The MEO in the MEC reference architecture is transformed into a MEC Application Orchestrator (MEAO). MEAO that uses NFV Orchestrator (NFVO), which is smart of the NFV MANO, for orchestration of resource and the set of MEC application VNFs as one or more NFV Network Services (NSs), [40]. By deploying the MEC in the NFV environment, low latency and scalability are the main benefits. NFV delivers scalability, and MEC offers low latency.

6.2.3 V2X Application

For vehicular communication, the automotive and telecommunication industries have invested in the connected car market as it supports digital transformation by exploiting communication technologies [190]. The V2X communications include the V2I, V2P, and V2N communications, which help to enhance vehicular traffic and safe driving and provide other services. The automotive industry supports acquiring MEC-based vertical segments in the 5G ecosystem. V2I communications require a low latency environment (less than 10 ms) [191]. MEC has been proposed as a possible solution for such stringent performance requirements, and it facilitates direct communication between nodes about V2X-related information via an underlying network. MEC technologies provide an ideal setting for use cases, allowing them to be linked entirely, communicate, and share information about traffic and maps. The application will also be able to stream videos and AR. For the V2X application, the MEC system provides cloud computing abilities at the network's edge, and its access-agnostic feature easily supports the deployment dependability via an underlying communication network.

6.2.4 Overview of the 5G-MEC V2X Testbed Elements

To develop and deploy a 5G-MEC tested for V2X application, the above mentioned elements need to be considered. 5G-MEC system delivers the support needed for the V2X application. Figure 6.5 refers to the real-time scenario that involves V2X communication and backend activities within 5G and MEC to satisfy the performance requirements. In addition to that, we investigated and denoted the generalized elements involved in building such testbeds. This segment provides an overview of the testbed and is discussed in two parts. The first one includes hardware components, and the other one includes software components.

Hardware elements

The hardware components are fundamental aspects of building the testbed. In terms of vehicular communication application, the testbed consists of multiple vehicles equipped with On Board Units (OBUs) with RSU connectivity. The vehicles can communicate with each other and the infrastructure via standard technologies. The RSUs are connected with the rest of the network using Ethernet links or industry standards. The end-user applications for the VUE are referred to as Client App that runs within the OBU. The OBU establishes a communication link with the RAN infrastructure via eNB, AP, or gNB. Each RAN infrastructure is equipped with MEH connected to the AP, eNB, and gNB. MEH may or may not be present at the RAN infrastructure. The RAN infrastructure can be provided by the MNOs or built using a Software-Defined Radio (SDR), such as USRP B210. MEH is the compute node resource for running the MEC applications and providing the services related to MEC. MEH is a high-performance computing unit. The transport network is the overlay network to provide connectivity between the RAN, MEC systems, and the data center/cloud. The data center/cloud can be private or public. In the case of the private cloud, high-performance servers are needed with or without GPU depending upon the orchestration responsibilities.

Software elements

The software elements represent the main virtualized components of the data plane. The application server/controller administrates and monitors the end-user application of VUE. An application server can be present in MEH or in a separate cluster. 4G and 5G CN represents the core network functionalities. The MEC system is deployed using different MEC building

blocks: MEP, MEPM, VIM, and MEO. MEP needs to be located within each MEH, whereas the rest of MEC elements has no constraint on their location. In addition, the testbed can include an SDN controller to perform network management and control the traffic flow. As presented by ETSI, MEC in the NFV environment offers NFVO for orchestrating the VNFs and VNFMs. The NFVO situates along with the MEO. As per our studies, the orchestration of MEC systems and VNF needs high-performance servers, and the allocation would be appropriate and suitable within the data center/cloud.

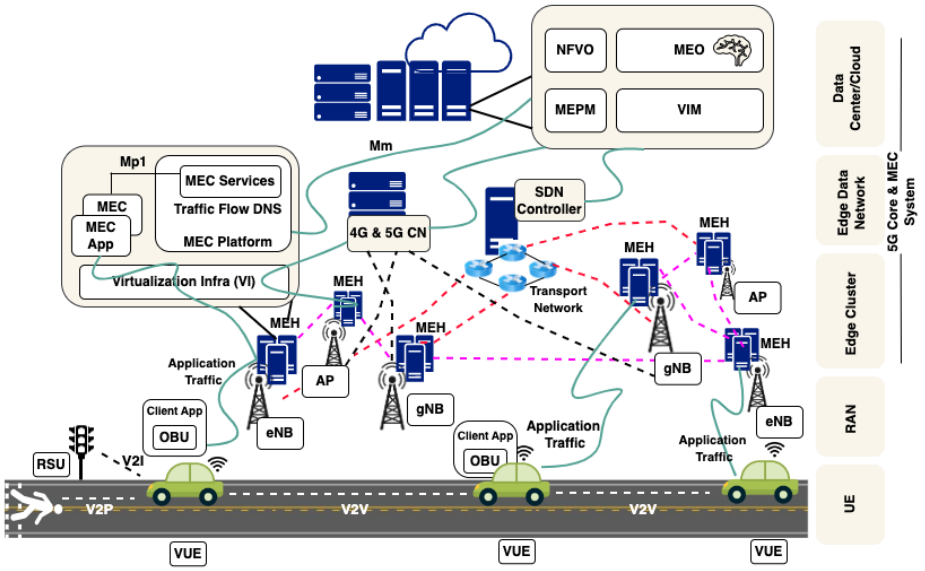


Figure 6.5: Overview of the 5G-MEC V2X Testbed Elements.

6.3 Existing Testbeds and Implementations

In this section, we focus on the existing and open-source activities on 5G-MEC testbeds for vehicular communications. The presentation is organized into three different subsections as follows: subsection 6.3.1 describes a set of important 5G-MEC testbeds, while subsection 6.3.2 is focused on 5G-V2X testbeds. Subsection 6.3.3 presents the features of the selected 5G-MEC V2X testbeds. We describe the deployment strategies and their respective use cases for each of the considered testbeds.

6.3.1 5G-MEC Testbeds

In the following, we present the selected 5G-MEC testbeds.

5GCity

5GCity testbed involves different platforms and services to support applications such as video streaming and smart city. 5GCity testbed provides MEC neutral host platform concentrating on Ultra-HD video streaming [192]. The deployment of the testbed is done in three European cities, Bristol, Lucca, and Barcelona. The project developed a 5GCity scalable orchestrator depending upon OpenStack as VIM and SDN RAN controller. MEAO is deployed using Open Source MANO (OSM) [55] and implements the Mm1 interface in a proprietary way [97]. It allows the creation and deletion of MEC applications and services. Furthermore, the functionalities of MEPM are performed. The MEC platform is deployed with an Mp1 interface to interact with the MEC applications. 5GCity Software Development Kit (SDK) supports the resource placement and the network slice application using the slice manager element. 5GCity SDK involves a monitoring manager for the VNFs and MEC applications [98].

Mosaic5G

Mosaic5G platform [99] provides flexible and scalable service deployment. The platform includes five software elements, such as OpenAirInterface (OAI) [41], for RAN and CN functionalities of the LTE network. FlexRAN [100] is an open-source implementation of Software Defined RAN (SD-RAN) deployed for controlling tasks within different base stations in a centralized manner or performing scattered control policies. Reconfiguring the control functionalities with RAN helps to acquire those tasks. Low Latency MEC (LL-MEC) is the MEC platform that separates the data plane and controls plane traffic at the network's edge. MEC functionalities and services are enabled using the LL-MEC platform. Mosaic5G deploys the NFV MANO platform using JOX to provide E2E network slices. Store involves modules to monitor and control applications.

5GINFIRE University of Bristol Testbed

The testbed of University of Bristol, developed within the 5GINFIRE project, is deployed for smart city use cases [101]. The testbed is developed

using open-source and licensed solutions. MEP and MEPM are deployed using the CloudBand of Nokia, which implements MEC functions and the VIM. The SDN architecture is based on the NetOS controller, while OSM is deployed for orchestrating the VNFs. Inter-Digital provides media services, including content distribution, which are utilized for the 5G smart tourism initiative. The media services are managed using OpenStack. The testbed involves mini-scale sensor nodes and equipment such as Raspberry Pi. The necessary data is acquired and processed at the data center.

EdgeX Foundry

EdgeX Foundry [48] is an open-source platform placed at the network's edge to build a MEC framework for IIoT applications. EdgeX is developed under the Linux Foundation Edge (LF Edge) project. At the code level, the golang version edgex-go of EdgeX Foundry has seven subdirectories: cmd, Api, Docker, Internal, pkg, snap, and bin. The cmd folder contains the project program entry `main.go` and describes key information such as the IP and port of the called EdgeX microservice; the Api folder contains the Application Programming Interface (API) in the form of HTTP for each microservice in the entire framework; the Docker folder contains instructions required to create the image; the Internal folder is responsible for the initialization of the device microservice driver and the processing of northbound command requests. Snap contains all EdgeX Go-based microservices; the bin contains executable files. To sum up, EdgeX Foundry provides a dual conversion engine: one is to uniformly convert data from different sensors and devices with various communication protocols and formats into common EdgeX data structures; the other is to specify customer-specific data through TCP/IP-based protocols format to provide data to applications, enterprises, cloud systems. EdgeX can run on small scale equipments such as Raspberry Pi [102].

6.3.2 5G V2X Testbeds

The 5G-V2X testbeds are essential to test V2X services without using the ETSI MEC architecture. For instance, these testbeds can be used to gauge the effectiveness of various edge and fog computing architectures alternative to the ETSI MEC architecture. Additionally, those testbeds enable the execution of experiments to assess the effectiveness of well-known communication systems in a real-world setting (for instance, network

effectiveness of vehicle communications at car speed in the street). The testbeds enable the testing of novel network protocols, for example, to reduce the power consumption on the radio access network or to increase the exchanged data reliability and/or security. Finally, the testbeds may also be utilized to test new technologies on the radio interface, such as mmWave, and the performance obtained by the cooperation among different Radio Access Technologies (RAT) in a multi-RAT scenario. Whereas the generic challenges of using 5G-V2X testbeds involve:

- The testbed/experiment complexity for troubleshooting and programming.
- Interoperability between MNOs/vendors for validating the end-to-end solutions.
- Real-world scenarios, such as network conditions and environmental factors, can impact testbed conditions.
- Security and safety issues of the connected vehicle and controlled environment to test the experimental analysis.
- Regulatory compliance with operating 5G V2X testbed.

The selected 5G-V2X testbeds are the following.

5GVINNI Munich Experimentation Facility Site

The 5GVINNI project [115] includes the 5GVINNI Munich testbed as the facility solution design. The testbed supports the eHealth use case named "Remote interventional support for emergency care application – mobile ultrasound." The testbed infrastructure support MEC functionalities, 5G RAN, and the 5G core of Huawei. The SDN part is based on the floodlight controller, while Huawei delivers the MANO and NFVI deployment. For the NFVI deployment, Docker container and Mininet are used. Mininet enables SDN capabilities that allow NFs inter-connectivity running on Docker through OpenvSwitch. The 5G site is connected to the core network using the Ethernet link for a high data rate. An ambulance is the UE where the 5G modem is mounted, which moves within the coverage area. The MEC functionalities are not fully deployed; Docker is used for the MEC services and applications deployment. MANO includes an NFVO that enables VNF instantiation and termination. The VNFM compiles a software module to initiate and manage the UPF Docker container [116].

5GINFIRE IT-Av Automotive Testbed

The 5GINFIRE project includes various testbeds, one of which is the "IT-AV automotive" environment, developed for experimenting with V2X applications in 5G systems. MEC-based solutions allowed the automotive sector to facilitate V2X communications and the exchange the vehicle-related information between nodes via the underlying communication network. In [82, 193], the authors present a novel V2X algorithm named Vulnerable Road User Safety (VRU-safe) that runs on the considered architecture. VRU-safe provides a low complex system, efficiency, and scope for identifying and predicting the hazardous events between vehicles and Vulnerable Road Users (VRUs). The performance and the result have been evaluated in the real-time 5G testbed of "IT-AV automotive".

5G V2X Testbed for Cooperative Automated Driving

The testbed is a 5G V2X wireless testbed built upon adjustable and reconfigured SDR and meant for cooperative autonomous driving [194]. The key novelty of this testbed is the Radio-Frequency (RF) front-end, which is re-configurable. In particular, the testbed allows testing new OFDM waveforms built on parameters such as pulse-shaping, a dynamic and self-contained structure layout, GNSS-assisted hybrid synchronization, and scheduled multiple access for reducing latency. The radio interface, which includes modulation, coding, frame structure, and multi-antenna systems, is freely designed and modified during the tests. The radio subsystem is simple to connect with the OBU inside the vehicle, which is a general-purpose computer-based system. The radio module can perform real-time analysis enabling live transmission, testing, data collection, and offline analysis for channel and automotive prediction. The radio module has a small form factor and power dissipation, making it ideal for automobile integration with constrained performance and power supply. For the RF Unit (RFU) setups, the radio subsystem construction components are NI/Ettus-based USRP X310 SDR platforms. A high-performance Intel hardware-based computer running a lightweight Linux operating system is utilized for the Baseband Unit (BBU). The RF external (RFE) subsystem is self-built and integrated with a power amplifier and switches based on Time Division Duplex (TDD) to increase transmitting power and reception sensitivity.

Two types of use cases are considered in the testbed: Type 1 use cases exhibit predictable behavior and are limited to three phases: planning, initialization, and execution; Type 2 usage scenarios are based upon emergency

braking and cooperative emergency maneuver. This testbed describes short-distance platooning and lane merging as the main Type 1 usage scenarios. Instead, Type 2 usage scenarios are distinguished by quick response times and the inability to plan ahead of time, typically occurring in emergency circumstances. The lab measurements and the field test carried out using this testbed show that the emergency brake trigger along with less than milliseconds latency [195].

6.3.3 5G-MEC V2X Testbeds

As concern 5G-MEC V2X scenario, we present the following testbeds.

5G-VINNI Moving Experimentation Facility Site

5G-VINNI project developed a moving experimentation facility site to support public safety communications, emergency response, and disaster relief applications. The moving experimentation facility site [117] accommodates the SATis5 testbed and uses Rapid Response Vehicle (RRV). The SATis5 testbed [118] includes an edge node and a central core network node for the satellite backhaul connection. At the same time, the UE connects to the edge node and central core network nodes via satellite backhaul. The backhaul is the transport layer for exchanging messages between the edge node and the central core network site. The Fraunhofer FOKUS Open5GCore toolkit has been deployed for the implementation of the 3GPP 5G core network. The Open5GCore toolkit provides the 3GPP Release 15 core network functionality and 5G New Radio integration. The testbed considers the NSA architecture where the traffic is split between the 4G and 5G at eNodeB. The testbed uses the 4G eNB for the RAN instead of 5G gNB for the base stations because the 5G gNB was not commercially available at that time. The Open5G core integrates into the iDirect satellite hub platform to operate the satellite network. The iDirect hub offers integration with the overall 5G architecture and SDN functionalities with the help of the Satellite Convergence (SatConv) layer, including the MNO for managing the modems and the satellite hub network. The iDirect Velocity Intelligent Gateway system introduces the 3GPP core network (SatCore) to the satellite hub platform.

The core network within the central node differs from SatCore and supports the satellite backhaul network. The satellite network operations are virtualized by moving their execution environment from a dedicated

server to a VM with the help of the OpenStack Pike VIM. Satellite VNFs incorporate the SatRAN software element, the satellite 3GPP Core Network function (SatCore), the satellite Network Management System (NMS), and auxiliary VNFs implemented on the same system that utilizes the OpenStack VIM. The iDirect Velocity Intelligent Gateway system is connected with the OSM regarding MANO. The SATis5 testbed [196] presents the SATCOM integration with the 3GPP 5G architecture for the over-the-air demonstration.

Smart Highway V2X Testbed

The Smart Highway V2X testbed [119] is a section of the CityLab testbed [197]. The Smart Highway V2X testbed, located in Antwerp (Belgium) is built on top of an open highway and comprises a highway strip with RSUs installed. These RSUs can handle V2X radio connections, such as short-range based on ITS-G5 and C-V2X with PC5 interface operating at 5.9 GHz and long-range C-V2X Uu, allowing them to communicate with vehicles equipped with such radio access technologies. 4G is used for long-distance connectivity. Experimenters can choose which communication modules to test on the RSUs, equipped with commercial and SDR modules. The RSU is fiber-connected to the virtualized cloud-based backbone. The Smart Highway V2X testbed [198] includes two mobile OBUs enabling V2X communications. The OBUs are also supported with long-range communications using 4G connectivity from Mobile Network Operator (MNO). The OBUs are equipped with the GNSS module for localization.

The Smart Highway V2X testbed provides the MEC collocation with RSU and tests the benefit of the intelligent application in a real-time environment [199]. For the MEC platform deployment and RSU, Kubernetes cluster deployment is done to monitor services based on Prometheus [61]. The real-time monitoring supports MEC applications and generates the statistics for containerized MEAO (cMEAO). The cMEAO provides the decision of the MEC host availability and location of the vehicles.

5G Carmen

The 5G Carmen testbed [120] involves integrating different MEC systems developed in three countries: Italy, Germany, and Austria. The MEC platform in the 5G Carmen testbed consists of three interfaces: connection to a Packet Gateway (PGW) via SGi-interface, public IP address exposed

to other MEC platforms, and VPN interfaces for platform management functions and applications deployed on the platform by the third parties. The setup in Germany includes the Deutsche Telekom integration and operation of the LTE/5G MEC system. The Nokia infrastructure provides the MEC platform, local edge processing capabilities, and the latency-optimized access path for LTE latency, reliability-sensitive applications. The MEC Edge cloud infrastructure is based upon the Nokia Airframe hardware, Ubuntu Linux and KVM virtualization, and the Application Life Cycle Management [200]. The setup in Germany deploys the BMW Cooperative Manoeuvring Application. TIM proposes the setup in Italy, and Nokia provides the MEC system solution and the AMQP broker platform provided by TIM. Some MEC platform and application entities are deployed in the virtual machines, while the remaining are realized in containers. A container-based service mechanism is used for life cycle management services. The application orchestration is provided via Helm charts. The VM orchestration in the MEC platform is done using Nokia CBAM. The setup in Austria is managed and operated by Magenta Telekom Austria (MTA) with the MEC environment in their network infrastructure. The setup is based on the OpenStack Red Hat Release 13 that can host multiple independent tenants. Currently, the setup is hosting the onboarding of MobiledgeX, which provides a low-latency computing platform.

Akraino Connected Vehicle

Akraino connected vehicle blueprint [64] focuses on the development of the MEC platform in specific toward the V2X application. Akraino Edge Stack [201] is an open-source software stack created by the Linux Foundation in February 2018 that supports access methods including 5G, LTE, Wireline, and Wi-Fi and provides high availability optimized for edge computing systems and applications cloud service. It is designed to improve the state of edge cloud infrastructure for enterprise edge, Over-The-Top (OTT) edge, and carrier edge networks, giving users a new level of flexibility to rapidly scale edge cloud services, maximizing the number of applications or users supported, helping to ensure the reliability of the system at runtime. The Akraino open-source project is organized around blueprints and uses blueprints to implement diverse edge computing-based use cases. Each blueprint contains the declarative configuration of the 3-layer stack described above, such as cloud platforms, APIs, and applications. Specifically, it includes the configuration statement, hardware, software, tools for managing

the entire stack, and delivery point (PoD, pod of delivery) of edge computing applications or architectures in a specific scenario. Each published blueprint enables automated edge infrastructure builds, edge stack site deployment, upstream and downstream project integration, automated orchestration, etc.

5G-DRIVE

The 5G-DRIVE testbed [121] focuses on developing and evaluating the eMBB and V2X scenarios. The network infrastructure built in the 5G-DRIVE testbed is based on Nokia's NetLeap LTE test network. In the 5G-DRIVE testbed, a virtual mobile network is enabled with its own EPC. The eNodeBs are interconnected with the OpenStack cloud and SDN-enabled backhaul. For the experimental tests, the Finland public site is used. Some of the research activities were focused on testing the coexistence of ITS-G5/LTE-V2X [202]. The 5G-DRIVE is equipped with the C-V2X box, a traffic light equipped with C-V2X RSU, and a tablet in the car. The field test involves the LTE network (2.6 GHz), 5.9 GHz RSU, and 5.9 GHz NEBULA OBU. The testbed has been used to experimentally evaluate the latency and the packet loss rate between the RSU and OBU [122].

Table 6.1 presents the categorization of the considered testbeds per their type and open-source accessibility. Furthermore, the equipment of VUE and RSU have mentioned in Table 6.1 for available 5G-V2X and 5G-MEC V2X testbeds. In the case of 5G-MEC testbeds, VUE and RSU are not applicable. In addition, Table 6.1 reports the sensors used in the surveyed testbeds for collecting the necessary data.

Table 6.2 presents the surveyed 5G-MEC and 5G-V2X testbeds based upon 5G usage scenarios categorization, such as eMBB, URLLC, and mMTC. Furthermore, the table briefly describes the surveyed testbed use case studies. The same information is summarized in Table 6.3 focused on 5G-MEC-V2X testbeds.

6.4 Usage Complexity of the Testbeds

The complexity of using the 5G-MEC V2X testbeds depends on the level of complexity required to set up, configure, and conduct the experiments.

Table 6.1: Summary on the considered testbeds, classified per type and open-source accessibility

Project/Title of the testbed	Type of testbed	Open source	VUE	RSU	Sensors	References
5GCity	5G-MEC testbed	Open source	Not applicable	Not applicable	Smartphone camera, CCTV camera and microphones	[97]
Mosaic5G	5G-MEC testbed	Open source	Not applicable	Not applicable	Wearable sensors for monitoring patient health-related data, specification is not provided, and LiDAR sensor	[203]
5GINFIRE University of Bristol testbed	5G-MEC testbed	Partially open source	Bike rider helmet, Raspberry PI, 360-degree Camera and audio	Not applicable	Camera, microphone, air quality, and noise sensors	[101]
EdgeX Foundry	MEC testbed	Open source	Not applicable	Not applicable	Video cameras, motion sensors and audio sensors	[48]
5GVINNI Munich experimentation facility site	5G V2X testbed	No	Ambulance	Huawei 5G RAN C-Band 3.41 GHz, Remote hospital integration	Ambulance, equipped with a HUAWEI 5G experimental UE, high-resolution camera and ultrasound system. Hospital is equipped with a display monitor, Camera, PC, and remote medical expert.	[115]
5GINFIRE IT-Av automotive testbed	5G V2X testbed	Partially open source	Bike and commercial car	Single board computer (SBC), DSRC wireless interface (IEEE 802.11p), WiFi interface (IEEE 802.11b/g/n), 4G Interface, GPS receiver, Antennas for each technology	OBUs internal sensors available (accelerometers, heading, speed, link quality connection, GPS, compass, RSSI, car neighbor's density, etc.)	[204]
5G V2X testbed for cooperative automated driving	5G V2X testbed	No	Commercial autonomous car	NI/Ettus USRP X310 SDR with Intel-based PC	LiDAR, Radar,GPS/IMU and CAN odometry	[194]
5GVINNI moving experimentation facility site	5G-MEC V2X testbed	No	SES's Rapid Response Vehicle (RRV)	Not applicable	Phone, a wearable, a car, and a fixed IoT sensor aggregator	[205]
Smart highway V2X testbed	5G-MEC V2X testbed	Partially open source	BMW X5 xDrive25d LO (2014) provided by University of Antwerp	Intel Xeon Broadwell E5-2620V4 SDR: Ettus USRP N310 ITS-G5: Cohda Wireless MK05 RSU LTE-V: Cohda Wireless MK6c EVK Mikrotik wAP LTE kit	Lidar sensor: Velodyne VLP-16, Camera and CAN sensor box: Raspberry Pi 3 B+ with can shield	[206]
5GCarmen	5G-MEC V2X testbed	No	BMW and CRF cars	Hardware not available	Vehicle and RSU sensors such as LiDAR, Camera, Radar, GPS	[120]
Akraino connected vehicle	5G-MEC V2X testbed	Partially open source	Tencent V2X business	Tencent V2X business	Camera, Radar, LiDAR, GPS, Accelerometers and Gyroscopes	[64]
5GDRIVE	5G-MEC V2X testbed	No	Automated passenger car called 'Marilyn,' equipped with sensing systems and ECUs needed for automated driving functions. Finland's VTT Technical Research Centre provides the car equipment used in the 5G-DRIVE testbed.	Cohda Wireless, Dual ITS-G5 DSRC radio GPS, ITS-G5, DSRC and WiFi (802.11abgn)	3 different 905 nm LiDARs, 24 GHz radar, Stereo camera, Inertia unit	[207]

Table 6.2: Surveyed testbeds use cases - 5G-MEC and 5G-V2X

Project/Title of the testbed	eMBB	URLLC	mMTC	Short description of case study	References
5GCity	Yes	No	Yes	5GCity project focuses on different use cases (UC). UC1 focuses on unauthorized waste dumping prevention. UC2 examines the platform capabilities using a single infrastructure to provide services to multiple MNOs. UC3 analyses and edit the real-time video acquisition of audience smartphones during an event.	[208]
Mosaic5G	Yes	Yes	Yes	Mosaic5G platform has been used for a few use cases, such as critical e-Health, V2N communication for intelligent transportation systems, and multi-service management and orchestration for smart cities.	[209]
5GINFIRE University of Bristol testbed	Yes	No	Yes	The 5GINFIRE University of Bristol testbed provides a smart city safety use case with capabilities that address daily needs ranging from parking and water treatment to city security.	[210]
EdgeX Foundry	Yes	No	Yes	The EdgeX platform has been used to build automation, video sensor security, and IoT control system use cases.	[102]
5GVINNI Munich experimentation facility site	Yes	Yes	No	5GVINNI Munich facility site used to perform the eHealth use case named "Remote interventional support for emergency care application–mobile ultrasound."	[211]
5GINFIRE IT-Av automotive testbed	Yes	No	Yes	The 5GINFIRE IT-Av automotive testbed includes three use cases: 5GCAGE, 5G driving trainer (5G-DT), and VRU-SAFE. The 5G-CAGE use case aims to implement a city safety solution, the 5G-DT system determines the safety and ecological driving behavior, and the VRU-Safe experiment concerns the safety of VRUs and the prevention of accidents.	[193]
5G V2X testbed for cooperative automated driving	No	No	No	Type 1 use cases exhibit predictable behavior and are limited to three phases: planning, initialization, and execution; Type 2 usage scenarios are based upon emergency braking and cooperative emergency maneuvering.	[194]

Table 6.3: Surveyed testbeds use cases - 5G-MEC-V2X

Project/Title of the testbed	eMBB	URLLC	mMTC	Short description of case study	References
5GVINNI moving experimentation facility site	Yes	No	Yes	Rapid Response Vehicle (RRV) is mainly used for governmental and Public Protection and Disaster Relief (PPDR) use cases. 5GVINNI integrates the SATis5 testbed to present the usage scenario, such as a connected vehicle.	[117]
Smart highway V2X testbed	No	Yes	No	Smart highway V2X testbed tests and validates ultra-fast connectivity between V2V and V2I communications. The testbed supports the data-intensive applications in the vehicle and at the RSU using the MEC support.	[119]
5GCarmen	Yes	No	Yes	5GCarmen use cases are cooperative maneuvering, situation awareness, video-streaming in the vehicle, and green driving.	[212]
Akraino connected vehicle	Yes	Yes	No	The Akraino connected vehicle proposed use cases include identifying and tracking the bad car, smart navigation, improving safe driving, and reducing traffic rule violations.	[64]
5GDRIVE	Yes	No	Yes	5GDRIVE testbed use cases focus on the V2I. UC1 is green light optimal speed advisory (GLOSA), in which the automated vehicle communicates with the infrastructure. UC2 is intersection safety that emulates the pedestrian crossing the intersection as the vehicle approaches the same path.	[207]

In the following, we report some common factors that can influence the complexity of using the testbed.

Hardware components: The complexity of using a testbed can be influenced by factors such as the complexity of its hardware components and the number of required features. For the 5G-MEC V2X testbed, the hardware components mainly consist of servers and computing units used to deploy the 5G-MEC and V2X application servers. In some cases, open-source tools can be used for deploying the 5G-MEC, while telecom companies can provide the necessary infrastructure.

Software components: The complexity of using software elements in a testbed is influenced by several factors, such as the development of libraries, APIs, and SDKs used for deployment. These factors can significantly impact the usage of testbeds. For the 5G-MEC deployment, both open-source and proprietary/licensed SDKs can be used. Additionally, simulators can be employed for implementing V2X applications and scenarios.

Configuration: The complexity of using testbeds can also be affected by the amount of configuration needed to set up the hardware components, including servers and computing units, according to the requirements of the experiments, and the SDKs to ensure that the testbed works correctly. In the case of open-source tools, the deployment's complexity may be influenced by the availability of user documentation and the user's expertise.

User interface: A user-friendly interface is essential for utilizing the resources of the 5G-MEC or 5G-MEC V2X open-source testbed. Several testbeds are available and accessible for conducting research and running experiments.

User documentation: The project deliverable that provides the hardware infrastructure used for the required use cases and the GitHub Read-me files for setting up the projects/required testbed repositories are examples of user documentation that significantly impacts the testbed replication or usage complexity.

User experience: User experience is mainly rated based on all components complexity and the technicality of the testbed. In general, the 5G-MEC testbed requires user expertise as well.

The usage complexity of the testbed can directly impact the accessibility and usability of the testbed for the researchers. Reducing the complexity of the testbed may increase the efficiency and effectiveness of further experiments/research.

The complexity of usage for available 5G-MEC, 5G-V2X and 5G-MEC-V2X testbeds is shown in the Table 6.4, Table 6.5 and Table 6.3, respectively. In the tables, we also summarized the aforementioned common factors.

6.5 Summary on Available Tools

Implementing a 5G-MEC testbed needs SDKs, open-source tools, and hardware requirements. Hardware components are crucial due to their computing power required by the SDKs and the applications. This section broadly discovers the open-source tools used to implement the 5G-MEC system. In the context of the V2X applications, available simulators are mentioned. The open-source tools and the SDKs are divided as per their roles in the 5G-MEC V2X testbed.

6.5.1 5G and 4G Systems

The available open-source tools can help to implement and deploy the 5G and 4G core functionalities. However, most analyzed testbeds used a commercial infrastructure where the product included the SDKs. Commercial infrastructure SDKs are proprietary and licensed. Some of the open-source tools and SDKs are still in progress. In the following, we present the most common open-source solutions and tools used in the analyzed testbeds.

OAI [41] provides the RAN and the CN implementation that can be used for deploying the 4G and partially 5G functionalities. OAI plans to deploy full standalone 3GPP-compliant 5G CN. The OAI CN implementation is tested with commercial RAN solutions such as Amarisoft gNB. Furthermore, in order to avoid the transmission of radio signals over licensed spectrum, OAI can be connected to the open source state-of-the-art 5G UE and RAN (gNodeB) simulator, UERANSIM [42]. OAI CN supports primary functionalities such as UE registration, service requests, and PDU session management. OAI can support multiple UE deployments.

Open5GCore [43] is developed by the Fraunhofer FOKUS to provide 3GPP Release 15 and 16 core network functionalities. Open5GCore runs on typical hardware components such as Raspberry Pi and computing servers using containers, pods, or virtualized environments. Open5GCore is licensed.

Table 6.4: Main features of the 5G-MEC testbeds

Project/Title of the testbed	Hardware components	Software components	Configuration	User interface	User documentation	User experience
5GCity	Hardware supports seven VMs with the specification of 2 vCPU, 4GB RAM, and 20GB VHDD. Along with this, each use case has different VM requirements.	OpenStack, Docker, Kubernetes, OSM, i2CAT SDN RAN Controller, and 5GCity SDKs	The configuration required to set up the software components and the 5GCity SDKs is complex. Individual element deployment is possible, but compiling and running experiments is difficult.	None	Project deliverable, GitHub repositories, installation documents, and release notes.	5GCity details testbed and use case deployments, but replicating the testbed is complex. The information for configuring the testbed is not provided in a particular iteration.
Mosaic5G	Computers:Gigabyte GB-BRi7-8550, SDRs Ettus:USRP B210, USRP B200-mini, Mobile phones:Google/Huawei Nexus 6P, Google Pixel 2.	JOX juju orchestrator, Low Latency MEC (LL-MEC) platform, FlexRAN master controller, OAI-RAN, OAI-CN	The open-source platforms and customized solutions configure the Mosaic5G ecosystem. Subscribing to the Mosaic5G ecosystem is accessible through GitLab.	None	GitLab repositories	The Mosaic5G ecosystem provides detailed information on the configuration of the open-source platforms and the hardware requirements, making it easy to replicate the testbed for conducting experiments.
5GINFIRE University of Bristol testbed	Nokia RAN and CN, Corsa DP2100, Edgecore AS4610, Dell Power Edge T630 Compute Servers(1TB RAM and 100TB HDD Storage)	Nokia CN, OpenDaylight, Nokia and CloudBand MEC, OpenStack and Nokia VIM, OSM	The 5GINFIRE University of Bristol testbed does not provide configuration details.	5GINFIRE provides a wiki to access the testbed resources for conducting experiments using the portal. Researchers can receive information on the available testbed providers and conduct necessary experimental analyses. However, access to the testbed resources is limited, and the 5GINFIRE portal is no longer operational. The testbed operates under different policies and is part of the smart internet lab at the University of Bristol.	For 5GINFIRE, user documentation is provided on both GitHub and the 5GINFIRE wiki portal. Additionally, project deliverables describe use cases based on the testbed.	The 5GINFIRE University of Bristol testbed does not provide configuration and setup information. Some elements of the testbed are deployed using open-source platforms, while others are proprietary. Replicating the testbed is challenging, and accessing resources is also difficult.
EdgeX Foundry	VM based on x86 or ARM architecture	Docker	EdegX is an open-source platform developed by the Linux foundation. The EdgeX Foundry ecosystem wiki provides information on configuration and installation.	None	GitHub README and GitHub Repositories	The EdgeX ecosystem is an open-source platform that offers a MEC solution for IoT and automation applications. It is built with Docker and used in various industrial applications.

Table 6.5: Main features of the 5G-V2X testbeds

Project/Title of the testbed	Hardware components	Software components	Configuration	User interface	User documentation	User experience
5G-VINNI Munich experimentation facility site	5G RAN (Huawei) 3.41 GHz, 5G Core (Huawei), MANO and NFVI (Huawei), SDN (Floodlight), Intel(R) Xeon(R) E5-2697 v2 @ 2.70GHz CPUs, 80GB RAM	Docker and Mininet	The configuration for operating the NFVI is limited to Docker and Mininet. Configuration for 5G RAN and Core is done by Huawei, and the ETSI-specified MEC architecture is not implemented. Docker virtualization technologies are used to support the edge site.	None	Project deliverable	The 5GVINNI Munich experimentation facility site testbed provides minimal information on the setup and experiment, making it difficult to replicate either.
5GINFIRE IT-Av automotive testbed	72 vCPUs, 704 GB of memory, 2 TB of storage, NIC Passthrough, DPDK and SR-IOV capabilities, 1x 24-port 1Gbps switch, interconnecting the infrastructure.	SUMO Simulator, OpenStack, OSM, OpenDaylight	The 5GINFIRE IT-Av automotive testbed does not provide configuration information. Along with utilizing open-source platforms, the testbed provides details on the SUMO simulator scenario.	Same as 5GINFIRE University of Bristol testbed. However, the IT-Av automotive testbed runs under the 5GASP project with similar objectives and is part of Aveiro Tech City Living Lab (ATCLL).	Same as 5GINFIRE University of Bristol testbed	The 5GINFIRE IT-Av automotive testbed does not include setup and configuration details. However, the testbed utilizes open-source platforms for experimentation, making replication possible.
5G V2X testbed for cooperative automated driving	NI/Ettus USRP X310 SDR with Intel-based PC	C/C++ with efficiency optimization based on Intel's x86 single instruction	None	None	None	5G V2X testbed for cooperative automated driving does not provide details on the software components and the configuration.

Table 6.6: Main features of the 5G-MEC-V2X testbeds

Project/Title of the testbed	Hardware components	Software components	Configuration	User interface	User documentation	User experience
5G-VINNI moving experimentation facility site	Lenovo™ ThinkCentre M920q Tiny Server Edge Node Fraunhofer FOKUS', SES's satellites, SES's RRV, Dell R630, – CISCO switch CATALYST 2960XR, MEC Server Super Server 5019A-FTN4	Open5GCore, Open Baton, OSM OpenStack, OpenNESS	Information regarding hardware and software components is provided for the testbed setup. While most software components are open-source platforms, the configuration of components per experiment is not provided. For end-to-end (E2E) orchestration, Fraunhofer FOKUS provides a solution using Open Baton and OSM.	None	Project deliverable	The 5GVINNI moving experimentation facility site is partially developed using open-source platforms. However, detailed deployment information regarding the platforms per experimental needs is not provided. Due to the lack of information, replicating the testbed is challenging.
Smart highway V2X testbed	General purpose computing unit (GPCU) Intel Xeon Broadwell E5-2620V4 @2.1GHz (8 Cores)32GB DDR4-2666 2Rx4 LP ECC	Docker, Kubernetes, Customized cMEAO	Smart highway V2X testbed is part of the Citylab testbed and offers comprehensive information on the software and hardware components.	Testbed resources are available instantly to conduct experiments. The FED4fire portal allows for registration and access to the resources, including the option to reserve them. Detailed information regarding accessing the testbed is available on the FED4fire portal.	The user manual is divided into various sections with detailed instructions. The jFed tool was created by FED4fire to access testbed resources. Using the guidance provided by the Citylab testbed, hardware components can be registered and reserved.	The Smart Highway V2X testbed is an easily accessible open-source testbed with a thorough procedure explanation that enables researchers to use the hardware and SDKs and conduct their experiments. The testbed allows for creating and deploying scenarios and access to all involved nodes, making it possible to repeat experiments or utilize the testbed resources.
5G Carmen	Intel NUC7i7DNBE, x86 server	OAI, KVM hypervisor, Docker, Kubernetes, Customized MEC platform and LightEDGE, OSM, and LightMANO	The configuration of the testbed is not provided in detail, although the elements are deployed using open-source platforms and customized solutions.	None	Project deliverable	The 5GCarmen testbed provides details about the testbed and use cases for conducting experiments, as described in the project deliverable that outlines the testbed connectivity and uses case flow. However, the testbed is currently unavailable and difficult to replicate.
Akraino connected vehicle	4 Arm/X86 server, MEC Platform(1 Server) + 1 App Server(1 Server)+ 2 Simulators(2 Server)	OpenStack, Kubernetes, Docker, DPDK, OpenNESS, OVS	Akraino Connected Vehicle provides a comprehensive blueprint of the testbed architecture and offers detailed documentation to access the testbed resources. This makes it easier for researchers to set up and replicate experiments in the connected vehicle domain.	Akraino connected vehicle provides access to the servers using SSH. To run the experiment, the required details are provided using Gerrit.	Gerrit Documentation, Akraino wiki	Akraino connected vehicle provide a hardware and software element list in the blueprint but do not offer support for the setup and configuration of the testbed. The testbed resources can be accessed through the portal, but replication may be challenging.
5G-DRIVE	Intel® Core™ i7-8700 CPU@ 3.2 GHz 16 Gb RAM 256 Gb HDD Intel Xeon Silver 42162.1GHz, 16C/32T, 9.6 GT/s32 GB RDIMM, 1 TB	OAI and Amarisoft, OpenStack, OSM, magma EPC	The setup and configuration of the 5GDRIVE testbed are not disclosed.	None	Project deliverable	The configuration information is unavailable, the 5GDRIVE testbed cannot be replicated, and the MEC deployment is inadequately explained. The testbed is not publicly accessible or open-source.

MagmaEPC [213] is developed within the Linux foundation projects. MagmaEPC is an open-source tool that supports extendable mobile core network functionalities. It can be deployed both using the Docker and on bare metal. MagmaEPC has three significant elements access gateway, orchestrator, and federation gateway.

NextEPC [214] is an open-source tool for implementing the 4G/5G CN and is not entirely compliant with 3GPP. NextEPC includes CN functionalities such as Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (SGW), Packet Data Network Gateway (PGW), and Policy and Charging Rules Functions (PCRF).

Open5GS [44] is an open-source implementation for 4G, 5G NSA CN, and 5G SA CN. Open5GS can be deployed using Docker and virtual machines. This CN implementation can be integrated with the open-source RAN simulator UERANSIM. Using UERANSIM, multiple UEs can be deployed and supported by the Open5GS CN.

Free5GC [45] has been developed mainly by National Chiao Tung University (NCTU). This implementation is based on the NextEPC platform, and the current version only supports the basic 5G CN, although it is planned to develop 5G CN functionalities defined in 3GPP Release 15 and beyond. Free5GC is deployed using virtual machines.

Simu5G [46] is an open-source simulator for 5G new radio (NR) based on SimuLTE libraries. Simu5G is developed using OMNET++ simulation framework. Simu5G features can allow testing resource allocation and management in a 5G network in addition to the targeted UE, depending on the modulation schemes. Simu5G is also tested with the ETSI MEC system model.

6.5.2 Networking

In the 5G-MEC system, networking is crucial for connecting the UE to the MEHs and MEC applications. Additionally, networking within the 5G CN to the MEC system needs to be established. The MEC orchestrator or MANO can have control over the 5G-MEC system via networking.

Data Plane Development Kit (DPDK) [215] is the data plane development kit developed by the Linux foundation. DPDK includes a collection of libraries for performing efficient packet tracing. DPDK performs microservices that support Container Network Interface (CNI) and is managed using a standard SDN controller.

Mininet [216] is an open-source network emulator used to create virtual networks that run on Kernel, switch, and application code. Mininet provides support for developing applications based on OpenFlow. Mininet hosts run the standard Linux software and can provide efficient routing along with SDN.

Open vSwitch (OVS) [217] is an open-source tool for the multi-layer virtual switch that provides redirection of networking traffic within virtual machines. Furthermore, OVS operates as a soft switch running in the hypervisor. Akraino connected vehicle implements OVS with OpenNESS to control the network traffic between containerized applications.

6.5.3 SDN

ETSI refers to virtualization technologies such as SDN as key enablers for network slicing applications and recommends considering it within its MEC architecture. SDN differentiates the control plane and user plane functionalities. The SDN controller increases network traffic efficiency by providing management and a global view of the network topology. A survey of SDN-based MEC solutions for network slicing and V2X URLLC applications is presented in [218], while Zhu et al. [219] investigated SDN controllers and classified their work based on application needs.

OpenDaylight [220] is one of the most popular open-source SDN controllers driven by the Linux foundation. The OpenDaylight SDN controller supports the OpenFlow southbound interface protocol. To test this controller, a switch is required, or an alternative solution such as Mininet can be used. OpenDaylight is based on a Java model controller that employs YANG as its modeling language. The configuration and usage of OpenDaylight are thoroughly documented.

Open Network Operating System (ONOS)[221] is an open-source tool developed using Bazel of Google. ONOS supports different southbound protocols, of which OpenFlow is the primary one. The ONOS SDN controller can be hosted on a VM and integrated with Mininet. ONOS provides comprehensive documentation and prerequisites for its usage.

Secci et al. present the performance comparison of OpenDaylight and ONOS SDN controllers [222].

6.5.4 MEP

The ETSI architecture can be customized to build a MEC system in which the MEP plays a pivotal role in delivering MEC applications hosted on MEH. The MEP provides the necessary MEC services for MEC applications. The MEC ecosystem comprises a list of ETSI-compliant MEC development tools, many of which are still in development. The implementations are classified based on the ETSI MEC-compliant architecture. MEH encompasses the MEP and MEC applications running on the VI.

The **5GCity SDKs** [97] MEC system builds a customized MEP on virtualized infrastructure compliant with the ETSI architecture. To deploy the 5GCity MEC system, VMs are required, and the specifications are listed in Table 6.4. The 5GCity MEP facilitates MEC application notifications and enables the creation, deletion, and discovery of new services required by the MEC system. Furthermore, the 5GCity MEP manages the DNS platform and DNS rules within the MEH. The 5GCity MEP can also communicate with third-party applications and manage the lifecycle of the MEC applications. The 5GCity MEC system implements the Mp1 and Mm5 interfaces in a proprietary manner.

The Low Latency MEC (**LL-MEC**) [51] platform is partially compliant with the ETSI MEC architecture. The LL-MEP aligns the Mp1 interface with the MEC application and the Mp2 interface with the virtual infrastructure to support low-latency MEC applications. The Mp1 interface uses a REST API to enable MEC services, while the Mp2 interface focuses on DNS rules and traffic routing. The LL-MEP also allows for the separation of the user plane from control plane functions using SDN principles and the OpenFlow protocol. Additionally, LL-MEP includes the RNIS MEC service standardized by ETSI for deriving RAN information. LL-MEP is mainly implemented using C++ on an x86 Linux system.

The **EdgeX foundry** [48] MEP is an industrialized solution developed by the Linux foundation. EdgeX is an open-source edge platform that is non-compliant with the ETSI architecture. It uses Docker for containerized MEC applications, and to manage these applications, EdgeX uses docker-compose running on an x86 Linux system. EdgeX supports virtual device services that simulate user devices, allowing for device and data management using the MQTT broker.

OpenNESS [47] is an Intel smart edge open-source MEC SDK complaint with ETSI architecture. It has an MEP and MEP manager that allows for building, deploying, and managing MEC applications. OpenNESS

includes different SDKs depending on the requirement, such as on-premises, access edge, and near-the-edge experience kits. It provides support for high-performance data planes and the management of virtual infrastructure. However, OpenNESS has two dissemination kits one is open-source, and the other is Intel-licensed distribution. OpenNESS has Intel-based hardware requirements to optimize the solution and run the experiments. In addition, it provides interfaces and the possibility to integrate with the orchestration VIM tools such as OSM, OpenBaton, and OpenStack, K8s. OpenNESS allows exploring resource allocation and management using edge multi-cluster orchestration (EMCO) flavor. It has an easy way of handling new application onboarding [223].

LightEDGE [49] is an open-source lightweight MEP solution compliant with the ETSI architecture. It includes the MEP manager, that is non-compliant with the ETSI architecture. LightEDGE utilizes microservices architecture and operates on a K8s cluster. The LightEDGE MEP includes several MEC services specified by ETSI, including the RNIS service. It also includes a UPF microservice, and 5G-EmPOWER is used to incorporate RNIS services to extract RAN information. LightEDGE [224] supports 4G and 5G environments and can integrate with the Open5GS and srsRAN solutions. LightEDGE is developed to integrate with OSM to have seamless orchestration.

AdvantEDGE [50] is a mobile edge emulation platform (MEEP) developed by Inter-digital. AdvantEDGE is compliant with the ETSI architecture that runs on Docker and K8s. It allows the sandbox creation to deploy the 5G-MEC scenario. AdvantEDGE provides ETSI-defined MEC services such as location, RNIS, WAIS, application enablement, application mobility, and V2X services. It also includes AdvantEDGE-specific edge services and network models to run the experiments. Furthermore, this platform allows controlling the network traffic between elements.

6.5.5 VI and VIM

The VI runs the MEP and applications by providing computation, memory, and storage capabilities. VI enhances MEC system deployment, and as a result, all open-source tools use virtualization. The VI manager is responsible for allocating, managing, and releasing the virtualized storage, memory, and computation as per real-time needs.

Docker [52] is a lightweight container-based virtualization tool mainly used for hosting MEC applications. However, some MEC tools used Docker

to deploy and run the MEP. Docker accelerates MEC application deployment and management with the help of Kubernetes (K8s). Early versions of the K8s cluster were based on Docker.

KVM hypervisor [225] is a Kernel-based virtual machine solution for Linux systems running on top x86 hardware. KVM allows the Kernel to perform as a hypervisor. 5GCarmen testbed uses a KVM hypervisor to host MEH and run the MEC applications.

LXC [226] is an acronym for Linux container. LXC runs and hosts a single Linux kernel system. LXC has experimented and tested for having low-level container run-time.

Kubernetes (K8s) [53] is an open-source tool to deploy, scale, and manage containerized applications. K8s helps to automate processes by creating a cluster to host multiple MEHs and allowing the initiation of containerized applications. It performs the role of a manager (VIM) for the container-based solution. K8s is used by many MEP tools and in the 5G-MEC testbed. Some of the 5G-MEC testbed solutions extend K8s usage for orchestration.

OpenStack [54] offers an IaaS cloud infrastructure solution to deploy and manage the VM. It is mainly used for larger-scale deployment models and controls large computing, storage, and networking resources. OpenStack allows configuring, creating, and deploying VMs based on specific requirements. However, in some cases, OpenStack is used to manage containers. Many MEC solutions do not use OpenStack; instead, K8s is preferred. ETSI refers to OpenStack for the VIM in the MEC system.

6.5.6 MEO and MANO

The MEO is responsible for various functionalities in the MEC system, such as tracking the overall view of the system, depending on the MEH deployment of computing, networking, and storage resources and MEC services. The MEO needs to ensure appropriate MEH selection and MEC application relocation as per user requirements. It also handles the onboarding of applications, packages, and SDKs. ETSI specifies the deployment model of the MEC system in an NFV environment, in which the MEC application and the NFV VNFs use the same VI. To fulfill the management and orchestration functionalities of the MEC system, ETSI has defined the NFV MANO architecture. Most of the surveyed testbeds deploy MANO functionalities using open-source tools and platforms.

OSM [55] is an open-source tool developed by the ETSI community. It delivers the MANO stack for NFV, capable of handling the VNFs and independent of VIM. However, OSM is compatible with open-source VIM, such as OpenStack. OSM has a minimum hardware resource requirement specification. OSM can deploy the MEC application as VNFs in the NFV environment. In this paper by Pablo Fondo-Ferreiro et al., OSM has been used to deploy the MEC orchestrator functionalities [146].

JOX [227] is a Juju-based service orchestrator and can interact with CN functionalities using the orchestrator plugins. JOX supports the orchestration of the virtualized network to deploy the network slices. JOX is compatible with the ETSI MANO NFV environment. JOX allocates network slices as per VNF and resource configuration specifications. JOX runs on juju VNFM and provides a plugin to the LL-MEP and FlexRAN SD-RAN controller.

OpenBaton [56] is an open-source platform providing the ETSI MANO functionalities. OpenBaton VNF packages are adaptable with Juju VNFM or generic VNFM. Generic VIMs can be integrated with the system via Juju VNFM. OpenBaton provides an NFV orchestration framework and manages MEC applications as VNF. OpenBaton is widely used for the deployment of network slice applications.

cMEAO [199] is a centralized MEC application orchestrator in the smart highway V2X testbed. cMEAO is compatible with the K8s VIM and its API. cMEAO allows allocating the resources at the network's edge by applying the smart mechanism. The smart mechanism does the application placement influenced by factors such as MEH availability, RSUs, latency requirements, and user geo-location. In addition to that, cMEAO supports the exchange of messages between the vehicles and with the edge.

LightMANO [57] is a lightweight open-source orchestrator platform compatible with the ETSI MANO architecture. LightMANO is built using Docker and runs within the K8s environment. LightMANO framework orchestrates the NFV services in a distributed system. LightMEP [228] adapts LightMANO as the orchestrator for its MEC system.

6.5.7 V2X

V2X application can benefit from the 5G-MEC testbeds and solutions to achieve URLLC. However, the V2X application deployment and integration with the MEC system still need to be completed. ETSI defines V2X

application services and its MEC deployment scenarios. V2X infrastructure is complex to run real-time testbed experiments. However, some work has achieved V2I communication, and some propose simulation tools to test and run the experiments as per the V2X application needs.

Simulation of urban mobility (SUMO)[59] is a primary open-source tool for creating and experimenting with the simulated traffic scenario. SUMO presents the real-world graph of road networks, where the vehicle has a unique identifier and information such as departure time and route. SUMO [229] includes a time-discrete solution where the default step length is 1 sec but can be reduced to 1 ms. Traffic and communication factors are essential to obtain and run the V2X simulated scenarios, and SUMO needs to be coupled with ns3. ns3 is an open-source simulator for external communication [60].

6.6 Discussion and Conclusion

5G-MEC solutions are proving to be fundamental in delivering high-latency applications with strict requirements and improving the user experience. MEC technology is a value-added service for MNOs and user application developers; MEC solutions are gaining commercial importance. However, there are research and practical development gaps between the ETSI-defined deployment model and the end solutions. The commercial or community solutions for the MEC system building blocks are incomplete, and some critical issues remain:

(i) How can the open-source 5G-MEC tools/platforms be integrated to create the 5G-MEC testbed or run experiments?

(ii) Are there any platforms to run the 5G-MEC V2X simulation/emulation?

(iii) Are the different MEC building block tools compatible with each other?

(iv) What APIs are available, and how can interfaces be developed between MEC elements?

(v) How can existing open-source 5G-MEC testbeds be used for experimenting with V2X applications?

In this paper, we examined existing 5G-MEC V2X testbeds and categorized them. Most 5G-MEC testbeds are built using open-source platforms and tools with specific hardware requirements. However, communities and commercial solutions are developed based on different 5G and MEC building blocks, as described in Section 6.5. Open-source communities are actively working to create support and develop API interfaces for integrating MEC elements. Standard VIMs can be integrated with most open-source MANO tools; however, MANO tools lack interface development for MEC platforms and MEC platform managers.

To facilitate the development and testing of 5G-MEC scenarios, ETSI has created a MEC sandbox solution that enables user application developers to experiment with the existing 5G-MEC environment. This sandbox solution simulates the 5G-MEC scenario and allows developers to gain a better understanding of the ETSI-defined MEC services.

The MEC system is divided into building blocks according to their functionalities. Scientific communities are working on each building block separately, which can shed light on issues such as compatibility. ETSI has defined the interfaces between the building blocks of the MEC system and presented their definitions and the REST API model. However, the development of the interface by the communities is still ongoing and needs to be completed.

In this paper, we have examined and presented the existing open-source 5G-MEC testbeds, some of which can be replicated, while others are accessible for running experiments. For replicating a 5G-MEC testbed for V2X application integration, one can use a SUMO simulator or access a smart highway V2X testbed.

The architectural approaches presented in [230],[231], [232] involve programmable edge-to-cloud virtualization, cloud-to-edge meta-operating system, and ML-assisted applications and aim to minimize the amount of data transferred between Cloud-Edge-IoT continuum, which is crucial for developing efficient and scalable systems. However, the surveyed testbeds must face several challenges for supporting and testing future data-intensive applications and IoT-based systems that run at the far edge. These testbeds offer limited storage capacities and finite processing capabilities of end-users and cannot easily integrate novel 5G-V2X wireless access solutions, such as NR-V2X, which are able to satisfy the required demanding features of automotive applications. In general, the 5G-MEC and 5G-MEC-V2X testbeds can provide a flexible platform and integration to try and enhance

the solution, which involves complexity and compatibility issues due to their adaptability and the need for more standardization. Finally, these testbeds do not natively integrate AI/ML, which is now mature enough to provide efficient solutions for complex optimization and prediction problems necessary to improve the orchestration and the network automation functions of beyond-5G systems.

Paper 2: Experimental Evaluation of Handover Strategies in 5G-MEC Scenario by using AdvantEDGE

Experimental Evaluation of Handover Strategies in 5G-MEC Scenario by using AdvantEDGE

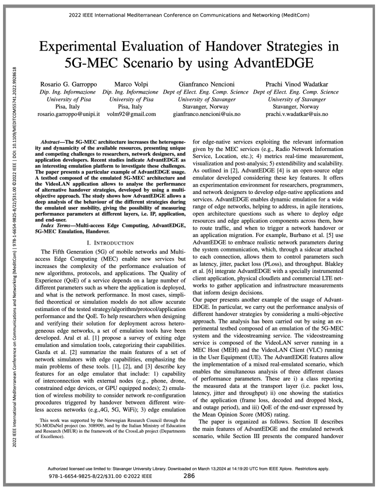
Rosario G. Garroppo[2], Marco Volpi[2], Gianfranco Nencioni[1], Prachi V. Wadatkar[1,2]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger

[2] Department of Information Engineering, University of Pisa

Published in the Proceedings of the Mediterranean Conference on Communications and Networking (MeditCom), 2022

https://ieeexplore.ieee.org/document/9928618



2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)

Experimental Evaluation of Handover Strategies in 5G-MEC Scenario by using AdvantEDGE

Rosario G. Garroppo, *Dip. Ing. Informazione*, *University of Pisa*, Pisa, Italy, rosario.garroppo@unipi.it

Marco Volpi, *Dip. Ing. Informazione*, *University of Pisa*, Pisa, Italy, volm92@gmail.com

Gianfranco Nencioni, *Dept of Elect. Eng. Comp. Science*, *University of Stavanger*, Stavanger, Norway, gianfranco.nencioni@uis.no

Prachi Vinod Wadatkar, *Dept of Elect. Eng. Comp. Science*, *University of Stavanger*, Stavanger, Norway, prachi.v.wadatkar@uis.no

2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom) | 978-1-6654-9825-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/MEDITCOM55741.2022.9928618

***Abstract*—The 5G-MEC architecture increases the heterogeneity and dynamicity of the available resources, presenting unique and competing challenges to researchers, network designers, and application developers. Recent studies indicate AdvantEDGE as an interesting emulation platform to investigate these challenges. The paper presents a particular example of AdvantEDGE usage. A testbed composed of the emulated 5G-MEC architecture and the VideoLAN application allows to analyse the performance of alternative handover strategies, developed by using a multi-objective approach. The study shows how AdvantEDGE allows a deep analysis of the behaviour of the different strategies during the emulated user mobility, giving the possibility of measuring performance parameters at different layers, i.e. IP, application, and end-user.**

***Index Terms*—Multi-access Edge Computing, AdvantEDGE, 5G-MEC Emulation, Handover.**

I. Introduction

The Fifth Generation (5G) of mobile networks and Multi-access Edge Computing (MEC) enable new services but increases the complexity of the performance evaluation of new algorithms, protocols, and applications. The Quality of Experience (QoE) of a service depends on a large number of different parameters such as where the application is deployed, and what is the network performance. In most cases, simplified theoretical or simulation models do not allow accurate estimation of the tested strategy/algorithm/protocol/application performance and the QoE. To help researchers when designing and verifying their solution for deployment across heterogeneous edge networks, a set of emulation tools have been developed. Aral et al. [1] propose a survey of exiting edge emulation and simulation tools, categorizing their capabilities. Gazda et al. [2] summarize the main features of a set of network simulators with edge capabilities, emphasizing the main problems of these tools. [1], [2], and [3] describe key features for an edge emulator that include: 1) capability of interconnection with external nodes (e.g., phone, drone, constrained edge devices, or GPU equipped nodes); 2) emulation of wireless mobility to consider network re-configuration procedures triggered by handover between different wireless access networks (e.g.,4G, 5G, WiFi); 3) edge emulation for edge-native services exploiting the relevant information given by the MEC services (e.g., Radio Network Information Service, Location, etc.); 4) metrics real-time measurement, visualization and post-analysis; 5) extendibility and scalability. As outlined in [2], AdvantEDGE [4] is an open-source edge emulator developed considering these key features. It offers an experimentation environment for researchers, programmers, and network designers to develop edge-native applications and services. AdvantEDGE enables dynamic emulation for a wide range of edge networks, helping to address, in agile iterations, open architecture questions such as where to deploy edge resources and edge application components across them, how to route traffic, and when to trigger a network handover or an application migration. For example, Burbano et al. [5] use AdvantEDGE to embrace realistic network parameters during the system communication, which, through a sidecar attached to each connection, allows them to control parameters such as latency, jitter, packet loss (PLoss), and throughput. Blakley et al. [6] integrate AdvantEDGE with a specially instrumented client application, physical cloudlets and commercial LTE networks to gather application and infrastructure measurements that inform design decisions.

Our paper presents another example of the usage of AdvantEDGE. In particular, we carry out the performance analysis of different handover strategies by considering a multi-objective approach. The analysis has been carried out by using an experimental testbed composed of an emulation of the 5G-MEC system and the videostreaming service. The videostreaming service is composed of the VideoLAN server running in a MEC Host (MEH) and the VideoLAN Client (VLC) running in the User Equipment (UE). The AdvantEDGE features allow the implementation of a mixed real-emulated scenario, which enables the simultaneous analysis of three different classes of performance parameters. These are i) a class reporting the measured data at the transport layer (i.e. packet loss, latency, jitter and throughput) ii) one showing the statistics of the application (frame loss, decoded and dropped block, and outage period), and iii) QoE of the end-user expressed by the Mean Opinion Score (MOS) rating.

The paper is organized as follows. Section II describes the main features of AdvantEDGE and the emulated network scenario, while Section III presents the compared handover

This work was supported by the Norwegian Research Council through the 5G-MODaNeI project (no. 308909), and by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence).

Abstract:
The 5G-MEC architecture increases the heterogeneity and dynamicity of the available resources, presenting unique and competing challenges to researchers, network designers, and application developers. Recent studies indicate AdvantEDGE as an interesting emulation platform to investigate these challenges. The paper presents a particular example of AdvantEDGE usage. A testbed composed of the emulated 5G-MEC architecture and the VideoLAN application allows to analyse the performance of alternative handover strategies, developed by using a multi-objective approach. The study shows how AdvantEDGE allows a deep analysis of the behaviour of the different strategies during the emulated user mobility, giving the possibility of measuring performance parameters at different layers, i.e. IP, application, and end-user.

7.1 Introduction

The Fifth Generation (5G) of mobile networks and Multi-access Edge Computing (MEC) enable new services but increases the complexity of the performance evaluation of new algorithms, protocols, and applications. The Quality of Experience (QoE) of a service depends on a large number of different parameters such as where the application is deployed, and what is the network performance. In most cases, simplified theoretical or simulation models do not allow accurate estimation of the tested strategy/algorithm/protocol/application performance and the QoE. To help researchers when designing and verifying their solution for deployment across heterogeneous edge networks, a set of emulation tools have been developed. Aral et al. [233] propose a survey of exiting edge emulation and simulation tools, categorizing their capabilities. Gazda et al. [234] summarize the main features of a set of network simulators with edge capabilities, emphasizing the main problems of these tools. [233], [234] , and [235] describe key features for an edge emulator that include: 1) capability of interconnection with external nodes (e.g., phone, drone, constrained edge devices, or GPU equipped nodes); 2) emulation of wireless mobility to consider network reconfiguration procedures triggered by handover between different wireless access networks (e.g.,4G, 5G, WiFi); 3) edge emulation for edge-native services exploiting the relevant information given by the MEC services (e.g., Radio Network Information Service, Location, etc.); 4) metrics real-time measurement, visualization and post-analysis; 5) extendibility and scalability. As outlined in [234], AdvantEDGE [236] is an open-source edge emulator developed considering these key features. It offers an experimentation environment for researchers, programmers, and network designers to develop edge-native applications and services. AdvantEDGE enables dynamic emulation for a wide range of edge networks, helping to address, in agile iterations, open architecture questions such as where to deploy edge resources and edge application components across them, how to route traffic, and when to trigger a network handover or an application migration. For example, Burbano et al. [237] use AdvantEDGE to embrace realistic network parameters during the system communication, which, through a sidecar attached to each connection, allows them to control parameters such as latency, jitter, packet loss (PLoss), and throughput. Blakley et al. [238] integrate AdvantEDGE with a specially instrumented client application, physical cloudlets and commercial LTE networks to gather application and infrastructure measurements that inform design decisions.

Our paper presents another example of the usage of AdvantEDGE. In particular, we carry out the performance analysis of different handover strategies by considering a multi-objective approach. The analysis has been carried out by using an experimental testbed composed of an emulation of the 5G-MEC system and the video streaming service. The video streaming service is composed of the VideoLAN server running in a MEC Host (MEH) and the VideoLAN Client (VLC) running in the User Equipment (UE). The AdvantEDGE features allow the implementation of a mixed real-emulated scenario, which enables the simultaneous analysis of three different classes of performance parameters. These are i) a class reporting the measured data at the transport layer (i.e. packet loss, latency, jitter and throughput) ii) one showing the statistics of the application (frame loss, decoded and dropped block, and outage period), and iii) QoE of the end-user expressed by the Mean Opinion Score (MOS) rating.

The paper is organized as follows. Section II describes the main features of AdvantEDGE and the emulated network scenario, while Section III presents the compared handover strategies. Section IV describes the testbed and the performance metrics, while Section V shows the results. Finally, Section VI concludes the paper.

7.2 Emulated Network Scenario

AdvantEDGE is Mobile Edge Emulation Platform (MEEP), enabling experimentation with edge computing. In particular, AdvantEDGE facilitates exploring MEC deployment models and their impact on applications and services in short and agile iterations [236]. The MEC APIs provided by AdvantEDGE allow to obtain information on the state of the network scenario. Furthermore, they allow changing the network characteristics and the location of devices on the network by sending API requests. AdvantEDGE allows the emulation of a tree network topology through which to forward packets to and from external services. Moreover, AdvantEDGE allows giving a physical position to the elements in the network, by inserting the geographical coordinates to each of them. By emulating the behavior of a connection, AdvantEDGE influences the data traffic flow with the impairments configured by the link pa-rameters (latency, jitter, PLoss, datarate). In order to simulate a dynamic scenario, AdvantEDGE allows the emulation of the client's movement.

In this study, the scenario consists of one client and one fixed MEH, located in Pisa near the Arno river. The UE can reach a MEH by using three alternative access network technologies: WiFi, 4G, and 5G. Obviously, depending on its geographical position, the UE can also be disconnected from every access network technology and, then, it cannot have access to the MEH services. The path of the UE is shown in Figure 7.1 using a blue line. In the same figure, the coverage of each network access technology is represented by different colors: WiFi in red, 5G in orange and 4G in blue. The Points of Access (PoAs) of the three technologies are co-located in two geographical points. The coverage radius of WiFi is 200 m, while 500 m and 1000 m is the coverage radius of 5G and 4G, respectively.

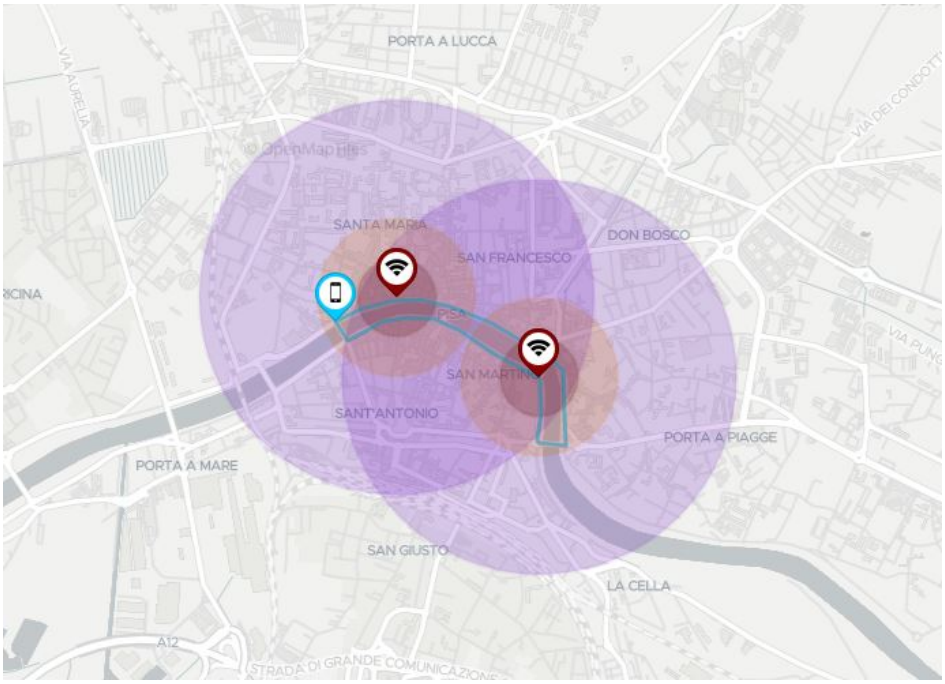


Figure 7.1: Map of the scenario considered in the experimental analysis with the AdvantEDGE platform

The AdvantEDGE representation of the network scenario is shown in Figure 7.2. The brown boxes are the applications, while the green is the physical UE. The antennas represent the PoAs. The MEC Application runs on the MEH(edge1 in the figure) connected to a point on the network called Zone3.

The Zone elements allow to group multiple network locations together. The logical Zone defines intra-zone network characteristics for traffic crossing between these network locations. In the figure, Operator1 is the Internet Service Provider (ISP) providing the IP connectivity through the three access technologies and the IP services supported by MEC.

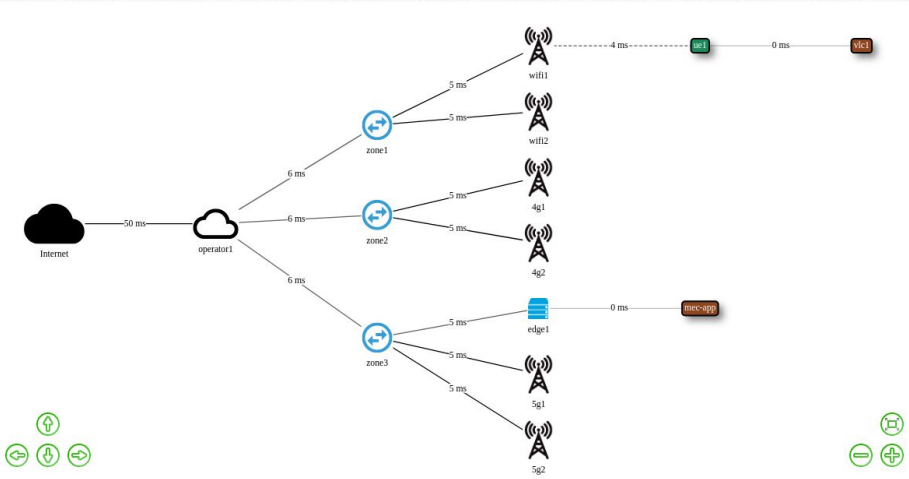


Figure 7.2: Network scenario described by the AdvantEDGE GUI.

The graph of the considered network scenario is shown in Figure 7.3. The attributes related to each link are respectively the PLoss probability, the jitter, and the latency. For each PoA, the figure shows also the available datarate.

Referring to Figure 7.3, Table 7.1 summarizes the value of each metric of the path between the MEH and the UE, as a function of the used PoA. The last row of the table shows the data rate (DR) available on the radio link in the congested scenario (C.S.).

In the study, the GIS API (getGeoDataByName) has been used to obtain information about the geographical position of the UE during its movements. These data allow estimating the distance between the UE and the different PoAs of the network. This information is used by the decision algorithm for establishing the PoA giving connectivity to the UE. The Sandbox API (sendEvent) of AdvantEDGE allows changing the PoA to which the UE is connected (i.e. perform the PoA handover) at runtime. The data necessary to build the graph, (i.e. arcs, nodes and the attributes value of the arcs) can be acquired runtime using the API of the MEC architecture. In order

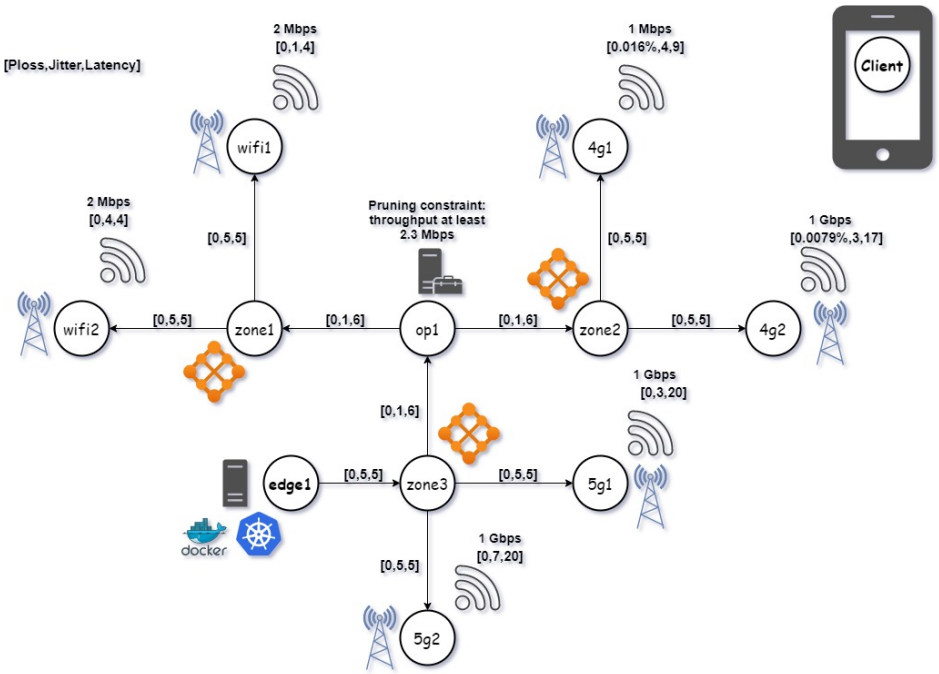


Figure 7.3: Graph of the considered network scenario.

to reduce the complexity of the experimental tests, the attributes values of the arcs are assumed to be constant during the experiments. Thus, these values have been manually configured in AdvantEDGE. The control plane procedure for performing handover between PoA of the same technology or between different technology, known as multi-Radio Access Technology (multi-RAT) handover, is not considered by AdvantEDGE. Hence, the delay and some performance issues (e.g., loss of packets or jitter increase) added by this procedure are neglected.

Table 7.1: Parameters of the Path for LDT and MDT Tests

MEH-UE	WiFi-1	4G-1	5G-1	WiFi-2	4G-2	5G-2
PLoss (%)	0	0.016	0	0	0.0079	0
Jitter (ms)	13	16	13	16	15	17
Latency (ms)	26	31	30	26	39	30
DR C.S. (Mbps)	2	1 129	1000	2	1000	1000

7.3 Compared Handover Strategies

The experimental analysis compares four different handover strategies, some of these use the following **Decision Algorithm**: the set of PoAs that can offer connectivity to the UE is ordered considering the values of PLoss, Jitter and Latency shown in Table 7.1. In particular, the strategy is to select the PoA with the lowest value of the PLoss metric. If more than one PoA has the same lowest PLoss value, the PoA with the lower jitter is chosen. In case these two steps output more than one PoA, the procedure considers the lowest value of the latency. Obviously, depending on the applications, the priority of the metrics used in the decision can be changed. The compared handover strategies are the following.

- **LTE-only with Default Throughput (LDT):** This case is the simplest one. The UE can connect only to 4G PoAs. The datarate available in each radio link is set to the default value, which is higher than the minimum datarate required by the application. In this test, the decision of the handover is simply performed evaluating when the UE is outside the coverage range of the serving PoA (i.e., 4G-1 or 4G-2). Then, the new PoA is the nearest one to the UE.

- **Multi-access scenario with Default Throughput (MDT):** In this case, the three available access technologies are considered. For each technology and for each PoA, the datarate is set to the default value. In other words, the DR C.S. values of Table 7.1 are not considered. Exploiting the GIS API information, for each UE position, runtime the set of PoAs able to guarantee the connectivity to UE is defined. This set is the input of the decision algorithm that outputs the selected PoA for the measured UE position. If the PoA returned by the algorithm is different from the serving PoA, the sendEvent of the Sandbox API is generated for performing the handover to the new PoA.

- **Multi-access scenario with Throughput Performance Data and Without pruning (MTPDW):** Differently from the MDT, in this case it is assumed that the radio links are congested. Then, for each PoA the available datarate is reported in the last row of Table 7.1. However, these values are not considered by the decision algorithm selecting the PoA offering the connectivity to UE.

- **Multi-access scenario with Throughput Performance Data and with Pruning (MTPDP):** This case is like the MTPDW, with the difference that the decision algorithm will consider the minimum datarate required by the application. In detail, the decision algorithm described above is applied on the subset of the PoAs, which is composed of PoAs providing a datarate higher than the minimum required by the application.

7.4 Description of the Testbed

The testbed used for the experimental analysis is described in Figure 7.4. Two physical PCs are used. The most powerful is based on a CPU Intel Core i7-8750H @ 2.20GHz, with 6 virtual CPU cores and 16 GB of RAM. This hardware has been configured to host the AdvantEDGE framework installed in a VirtualBox Virtual Machine (VM) running Linux Ubuntu 18.04 OS hosted by Windows OS. The VM has IP address 192.168.178.145. The same hardware is used as MEH with IP address 192.168.178.200. The MEH supports the entertainment video service, implemented by means of a VideoLAN server [239] running on the Windows OS. The VLC runs in a PC with Windows OS and has IP address 192.168.178.100. Since AdvantEDGE runs on the same PC implementing the MEH, the traffic forwarding is obtained referring to the port number. As reported in Figure 7.4, the traffic from the external to AdvantEDGE is addressed to the port number 30171, while 30141 is the port number used to forward the traffic from AdvantEDGE to the external node (i.e. the VLC). The MEH streams the video using the MPEG Transport Stream (MPEG TS) protocol, defined in the ISO/IEC standard 138181 [240], over UDP. The traffic is modified by AdvantEDGE depending on the network characteristics set on the scenario. As a consequence, the quality of the streamed video might be affected at various degrees. The chosen video used during the tests shows nature landscapes and lasts 334 seconds. Other features of the video are summarized in Table 7.2.

AdvantEDGE uses Grafana dashboard [241] to acquire data during the emulation. In particular, Grafana allows the visualization of different network statistics. Among these, a key statistic is the instant when the handover events happen. The data visualized by Grafana can be exported in csv format at the end of the experiment.

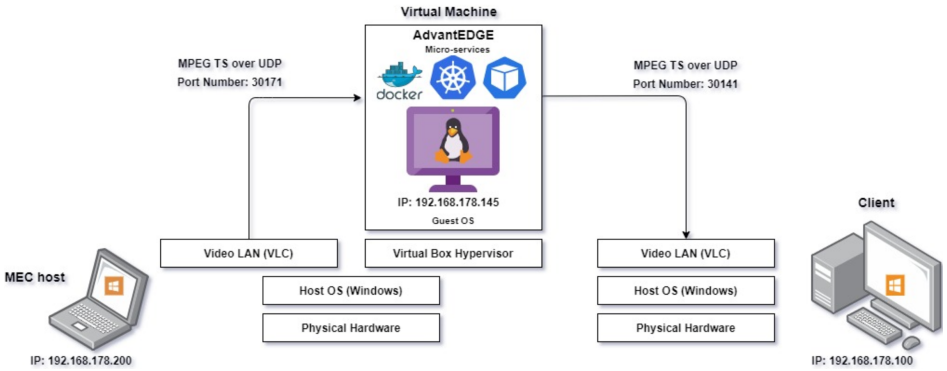


Figure 7.4: The testbed.

7.4.1 Performance Metrics

During each test, three different classes of performance parameters have been collected and analyzed. The first class refers to the set of parameters that are given by AdvantEDGE. These are the following:

- **Packet Loss (PLoss)**, defined as the ratio between the number of packets that fail to reach the egress point of AdvantEDGE (denoted as N_E) and the number of packets observed in its ingress point, N_I:

$$Ploss = \frac{N_I - N_E}{N_I}$$

  These losses are measured in the AdvantEDGE environment. For each packet and in each link of the network, the packet lost is randomly established with the probability set by the AdvantEDGE user. The observed values do not consider the losses in the link between the external nodes (i.e. MEH and VLC). In the testbed, these losses are zero, because a dedicated 1 Gbps Switched Ethernet LAN is used.

- **Latency**, defined as the time a packet takes to be transferred between the ingress and the egress point of AdvantEDGE. Considering a path

Table 7.2: Features of Transmitted Video

Codec	Video bitrate	Audio bitrate	Width	Height
H.264	2000 kbps	128 kbps	1280 px	720 px

composed of more than one arc, the overall latency is the sum of each latency value on the arcs. The AdavantEDGE generates the value of the latency of a packet in each link using a random variable with a Gaussian distribution, where its mean represents the latency value and its standard variation the jitter, provided as configuration parameters for each link.

- **Jitter** represents the measured standard variation of latency.

- **Throughput** is the maximum amount of data that can be transmitted in one second. In AdvantEDGE, a nominal datarate can be set for each link of the emulated network scenario. The reported value instead is measured by observing the selected traffic flow.

The second class refers to the subjective QoE that is observed by the end user during the service. The considered parameter is the MOS, which represents the mean of the absolute score given by the customers according to their satisfaction during the visualization of the video. As recommended by the ITU-T P800 standard [242], an Absolute Category Rating (ACR) is used to score the experience by using a five-point category-judgement, from 1 (Bad) to 5 (Excellent).

The last class contains a set of parameters given by the VLC. These parameters show the quality of the data transmission between the VideoLAN server and the VLC. The considered parameters are the following:

- **Decoded blocks** represent the number of encoded blocks that the VLC decoder converted into an uncompressed format.

- **Dropped blocks** are the number of dropped blocks. A drop can occur when the received blocks are not synchronized among them due to a delay in a network, or when a packet containing information about the video stream is lost.

- **Lost frames** refer to the number of lost frames during the reproduction of the streaming. These losses may occur when a block is lost or when the video decoder is unable to decode blocks.

- **Outage period** is defined as the amount of time during which the received video is stuck on the screen.

7.4.2 Preliminary Tests

The preliminary analysis is devoted to establishing the performance obtained in the best condition and to detecting the presence of some transient periods. During these preliminary tests, some anomalies were always noticed in the first 17 s of the video service. These anomalies can be related to the time required by the PC hosting AdvantEDGE to overcome the overloading faced during the set up of AdvantEDGE. Indeed, AdvantEDGE requires the running of different virtualized software modules connected to each other, which represent a heavy requirement for the hosting PC resources. Thus, all the shown performance results are collected by neglecting the first 17 s of the experiment.

The reference performance is obtained by running an experiment in ideal conditions. During this test, the UE is connected to a 5G PoA without movement. No losses, no jitter and no delay are configured in the paths of AdvantEDGE. The communication between the VideoLAN server of the MEH and the VLC is ideal, i.e. without packet loss, and with negligible jitter and delay, both added by the connection outside AdvantEDGE. In this ideal condition, the VLC graphical interface showed 19749 decoded blocks and 9903 displayed frames.

7.5 Measurement Results

7.5.1 Strategies with No Constraints on Throughput

These two experiments refer to the case where each PoAs can offer the maximum datarate, i.e. the last row of Table 7.1 is not considered. In other words, the assumption is that the datarate of each link is higher than the traffic rate generated by the VideoLAN server.

LDT strategy:

Figure 7.5 shows the observed latency for each packet, and its average values estimated over a moving window of 20 samples, when the simple LDT strategy is applied. As described in the legend, the vertical lines show the handover performed by the UE. The handovers can be deduced from the color change of the vertical lines.

The red vertical bars of the figure indicate the observed outage period during the experiment. The figure shows that the outage periods are more

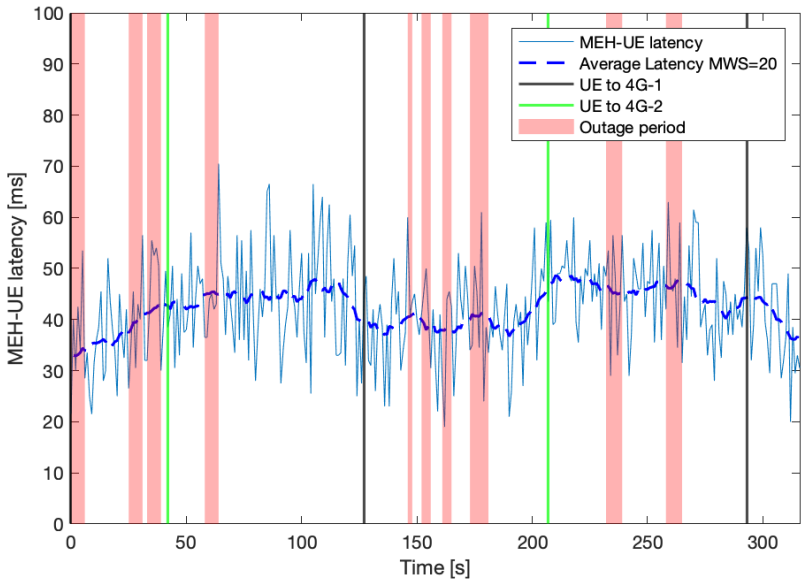


Figure 7.5: Latency and Handover events - Case LDT

frequent when the UE is connected to the PoA 4G-1. Indeed, this PoA has a packet loss probability higher than that of the PoA 4G-2 (see Table 7.1).

Table 7.3 summarizes the observed performance parameters.

Table 7.3: Observed Performance Parameters - Case LDT

Decoded Blocks	16133	**Outage period** (%)	11.98
Displayed Frames	8725	**Outage period 4G-1 (%)**	26.03
Lost Frames	1178	**Outage period 4G-2 (%)**	11.07
MoS	2	**Average latency (ms)**	42.36

The percentage of the outage period corresponds to 38 s, while the observed average length of the outage periods is 5.8 seconds. These periods negatively influenced the QoE, as shown by a MOS value equal to 2. The periods related to the two PoAs are calculated as the ratio between the observed outage period when the UE is connected to 4G-1/4G-2 and the total time of connection with PoA 4G-1/4G-2.

MDT Strategy:

The results obtained with the MDT strategy are summarized in Figure 7.6 During the experiment, no outage period has been observed. The quality of the video was high without any disturbing interruption. These results can be easily explained. Indeed, the only interruptions that could occur might be caused by the overloading of the MEH PC (due to limited resources), or during the brief intervals of the connection between the UE and the 4G-2 PoA, which has the highest packet loss probability. During this test, 23 handover events were observed, compared to 4 handover events of LDT. This observation shows that the decision algorithm solve the issues related to bad quality of connection, activating handover between PoAs. However, this approach could cause problems if the handover between the different PoAs takes a too long time (more than the video buffer time).

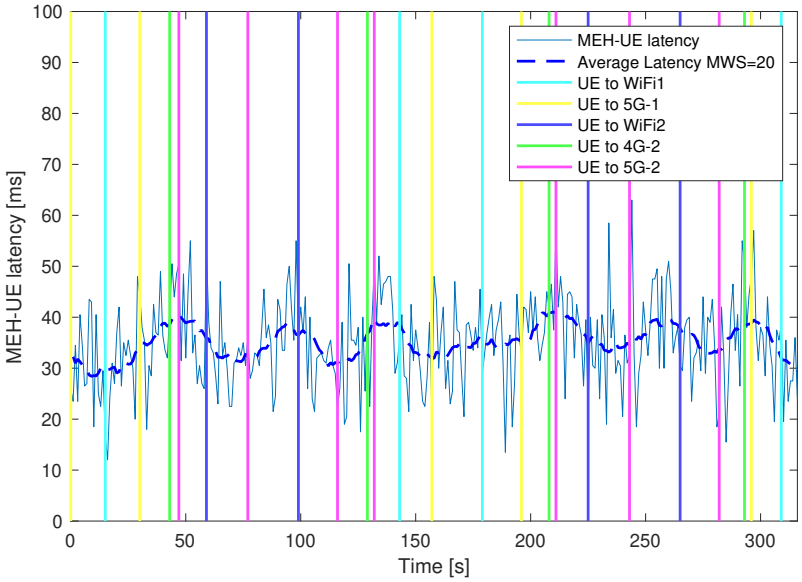


Figure 7.6: Latency and Handover events - Case MDT

After the transient period represented by the first 17 s of the test, no lost frame is observed. During the transient period, 176 frames have been lost. The average latency is 35.15 ms.

7.5.2 Strategies with Constraints on Throughput

With respect to the previous cases, in these experiments the datarate of the link between each PoA and the UE is set to the values reported in the last row of Table 7.1. Hence, some PoAs do not guarantee a datarate higher than the average video bitrate, which is equal to 2.32 Mbps.

MTPDW Strategy:

During this experiment, the decision algorithm is applied to the whole set of PoAs able to guarantee the connectivity to UE. No selection of the PoAs subset offering a minimum throughput is performed. Hence, the decision algorithm could choose a PoAs that does not satisfy the throughput requirements of the video service. This scenario leads to outage periods and also to events where a low quality of the video is observed. As shown in Table 7.1, the PoAs that do not satisfy the throughput requirements WiFi-1, 4G-1 and WiFi-2. The paths using these PoAs fail to guarantee sufficient throughput for forwarding the video traffic at the streaming bitrate speed.] Figure 7.7 shows the obtained results in terms of measured throughput. In the figure, the vertical lines with different colors represent the handovers between PoAs. The dashed red lines give the reference value of the traffic generated by the VLC application. This value has been obtained by measuring the traffic throughput in ideal network conditions, i.e. with no loss and very high datarate in network link. In the figure, the outage periods are represented by the red bars, while the low quality periods are represented by the blue bars.

Figure 7.7 clearly points out the correlation between the quality of received video with the measured throughput. Indeed, when the UE is connected with one of the three PoAs with a datarate lower than the required datarate, low quality events and/or an outage period occur. The playout buffer of VLC is set to 1 s. When this buffer is emptied and the new contents arrive too slowly, due to the insufficient throughput guaranteed by the path, an outage period occurs. VLC shows the contents, as soon as the buffer is filled up again. Obviously, when the throughput is insufficient, the time it takes to fill up the buffer is longer than the time it takes to empty it. Hence, the video is not fluid and looks like a set of pictures, showing a low quality.

In the figure, the peaks of the measured throughput (black line) are related to buffered traffic in the network that is delivered to the UE as

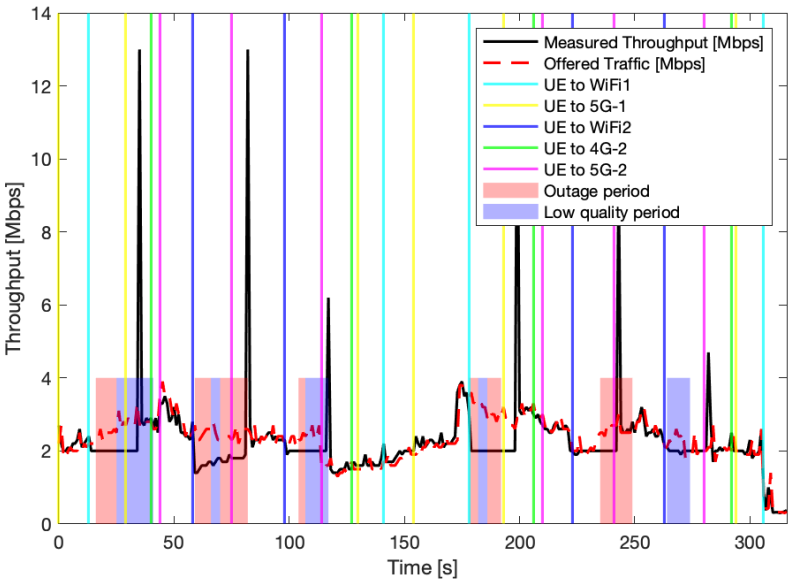


Figure 7.7: Bitrate, Handover events and Low Quality events - Case MTPDW

soon as the UE connects to a PoA with enough throughput. The other performance parameters as summarized in Table 7.4.

MTPDP Strategy:

In this experiment, the input of the decision algorithm is the subset of PoAs having a datarate higher than 2.32 Mbps (i.e., the measure average throughput of the video application), i.e. 5G-1, 5G-2 and 4G-2. Figure 7.8 shows the results obtained with the MTPDP strategy. The figure clearly points out that there is no outage period and not even low quality events.

Table 7.4: Observed Performance Parameters - Case MTPDW

Decoded Blocks	15363	**Outage period (s)**	55
Displayed Frames	7925	**Outage period (%)**	13.88
Lost Frames	1978	**Low Quality Period (s)**	44
MoS	1	**Low Quality Period (%)**	17.35

The throughput of the selected paths is always higher the requirements of the video service. The quality of the video is high, therefore the MOS score is 5. Except for the frames lost during the transient phase, no losses have been reported by the VLC statistics.

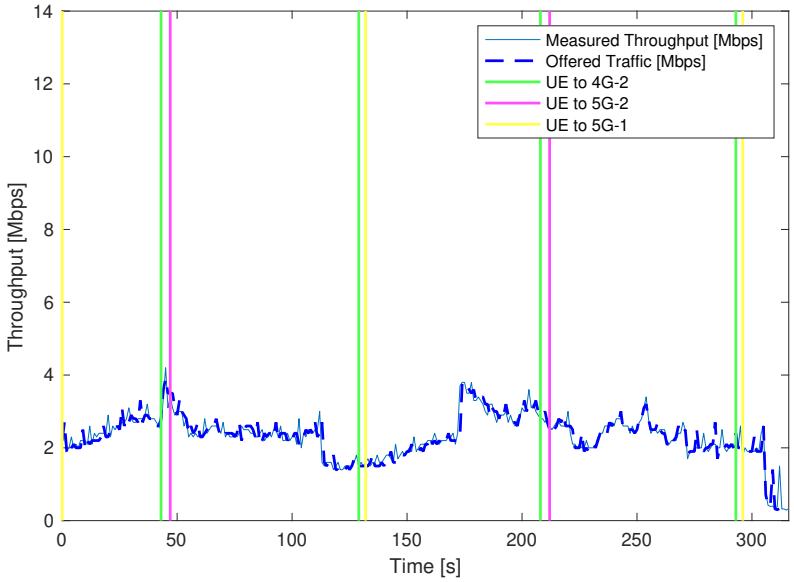


Figure 7.8: Handover events, Measured Throughput and Offered Traffic - Case MTPDP

7.6 Conclusion

The presented experimental analysis shows the large set of data that can be acquired with AdvantEDGE emulator interconnected with external devices. The presented analysis points out the performance enhancements given by the multi-objective strategy that considers the minimum throughput guarantee. As shown by the results, the QoE is maintained during the movement of the UE or the degradation of the network conditions.

Paper 3:
MEC Application Migration by using AdvantEDGE

MEC Application Migration by using AdvantEDGE

Prachi V. Wadatkar[1,2], Rosario G. Garroppo[2], Gianfranco Nencioni[1]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger

[2] Department of Information Engineering, University of Pisa

Published in the Proceedings of the 17th EAI International Conference on Tools for Design, Implementation and Verification of Emerging Information Technologies (TRIDENTCOM), 2022

https://link.springer.com/chapter/10.1007/978-3-031-33458-0_8

MEC Application Migration by Using AdvantEDGE

Prachi Vinod Wadatkar[1,2(B)], Rosario G. Garroppo[2], and Gianfranco Nencioni[1]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger, Kjell Arholms gate 41, 4021 Stavanger, Norway {prachi.v.wadatkar,gianfranco.nencioni}@uis.no
[2] Department of Information Engineering, University of Pisa, Via Girolamo Caruso, 16, 56122 Pisa, Italy rosario.garroppo@unipi.it

Abstract. Multi-access Edge Computing (MEC) and 5G are key technologies for the development of new applications requiring low latency and for computation off-loading. Emulation tools, such as AdvantEDGE, allow to rapidly test new services and resource management techniques in the 5G-MEC infrastructure. The paper presents an experimental study aimed to show the usage of AdvantEDGE tool for evaluating the migration performance of a MEC application. The key aspect of the study is that the application mobility is obtained by using the migration of the Kubernetes (K8s) application pod. The standard K8s does not have the ability to support the pod migration in a cluster of nodes. While recent research proposes a mechanism to migrate pod, there is no work investigating the migration technique with the AdvantEDGE MEC solution. Referring to a video service, the paper shows a scheme developed during the experimental study to allow the pod migration when K8s is used with AdvantEDGE. Using the emulation of user mobility given by AdvantEDGE platform, the described experimental tests allow to show the performance of the MEC application migration.

Keywords: Multi-Access Edge Computing (MEC) · AdvantEDGE · 5G-MEC Emulation · Kubernetes · Migration

1 Introduction

Due to the increasing demand for computational and Internet of Things (IoT) applications, the fifth generation (5G) of mobile networks will face unusual traffic volume. At end-users, computing-intensive applications become an inherent concern due to the end user's limited storage and computational abilities. Multi-access Edge Computing (MEC) is the emerging technology in the 5G network that can process a large amount of data within the Radio Access Network

This work was supported by the Norwegian Research Council through the 5G-MODaNet project (no. 308909).

S. Ya et al. (Eds.): TridentCom 2022, LNICST 466, pp. 104–119, 2023.
https://doi.org/10.1007/978-3-031-33458-0_8

Abstract:
Multi-access Edge Computing (MEC) and 5G are key technologies for the development of new applications requiring low latency and for computation off-loading. Emulation tools, such as AdvantEDGE, allow to rapidly test new services and resource management techniques in the 5G-MEC infrastructure. The paper presents an experimental study aimed to show the usage of AdvantEDGE tool for evaluating the migration performance of a MEC application. The key aspect of the study is that the application mobility is obtained by using the migration of the Kubernetes (K8s) application pod. The standard K8s does not have the ability to support the pod migration in a cluster of nodes. While recent research proposes a mechanism to migrate pod, there is no work investigating the migration technique with the AdvantEDGE MEC solution. Referring to a video service, the paper shows a scheme developed during the experimental study to allow the pod migration when K8s is used with AdvantEDGE. Using the emulation of user mobility given by AdvantEDGE platform, the described experimental tests allow to show the performance of the MEC application migration.

8.1 Introduction

Due to the increasing demand for computational and Internet of Things (IoT) applications, the fifth generation (5G) of mobile networks will face unusual traffic volume. At end-users, computing-intensive applications become an inherent concern due to the end user's limited storage and computational abilities. Multi-access Edge Computing (MEC) is the emerging technology in the 5G network that can process a large amount of data within the Radio Access Network (RAN) [243]. European Telecommunications Standards Institute (ETSI) provides the MEC specifications to meet the requirements of the applications where real-time processing is needed. The core idea of MEC is to deploy the cloud computing capabilities within the RAN, close to the end-user [17]. ETSI MEC Ecosystem [244] refers to the MEC solutions that support the experimentation and deployment of the practical scenarios that include a 5G-MEC framework. In those MEC solutions, a recent study [245–249] shows AdvantEDGE as a potential emulation platform tool to perform the different challenges in the MEC framework.

AdvantEdge [50] is a Mobile Edge Emulation Platform (MEEP) that runs on Kubernetes (K8s) [53] and Docker [52]. The emulation platform enables the analysis with edge computing technologies, applications, and services. AdvantEdge provides the ability to explore edge deployment models, and allows the user to modify the deployment scenarios considering elements such as network topology, network characteristics, application mobility, and UE movement. AdvantEdge provides the connection of real cloudlet and UE applications so that simulation can capture the impact of network design on application performance. AdvantEdge also allows the measurements collection in InfluxDB time series [250]. InfluxDB is a time series database built specifically for storing time series data.

Virtualization technologies support the deployment and management of the MEC applications and the MEC host (MEH). K8s is developed by Google [251] and is a superior technology for automating the management, scalability, and deployment of containers and nodes. Containers are prevalently used for running stateful applications. However, the standard K8s don't have an in-built mechanism to migrate the stateful containers from one node to another.

The main contribution of the paper is the description of an experimental testbed aimed at evaluating the performance of MEHs migration strategies using the 5G-MEC emulation scenario implemented by the AdvantEDGE

platform. Then, referring to a video streaming application, the paper presents an experimental analysis of the application migration in the runtime mode. The different time-related parameters related to the migration process are presented and analysed.

The paper is organized as follows. Section 8.2 gives the ETSI MEC application migration use case and the standardized ETSI MEC API for the application mobility using the AdvantEDGE platform. Section 8.3 presents the related work, while Section 8.4 discusses the main advantages of AdvantEDGE and the emulated network scenario considered for the experimental tests. Section 8.5 shows the testbed setup, the integration of the application migration techniques, and the working strategies. Section 8.6 evaluates the average values and confidence interval (CI) over the observation period of selected time-related performance parameters. The future work and the conclusions are summarized in Section 8.7.

8.2 Background

ETSI has specified the management of the MEC by considering the system level, the host level and the network layer functionalities. MEC Orchestrator (MEO) is the brain and has the overall view of the MEC system level management elements. The MEC system level consists of the MEHs, physical resources, applications and its services along with system topology. The MEO is the responsible entity for selecting the MEH during the application instantiation for the end user. In the ETSI MEC architecture, a MEH has a MEC Platform (MEP), which can establish a connectivity with the the other MEPs by using the Mp3 reference point. Mp3 reference point is the platform-to-platform interface that exchanges the information related to the application mobility between MEHs. In a disturbed deployment of the MEC system, multiple instances of the MEC application can be present and maintain the connectivity over different MEHs. The entities in the MEC application mobility within the intra-MEC system scope are presented in [81].

For the MEC application mobility, there are two entities to focus on: the application availability in the target host and the user context transfer. In the first entity, the application is required to be available in the targeted MEH, where the targeted MEH does not have designated application to provide the service to the end user. The MEO decides the application instantiation on the targeted MEH and has the ability to download the

application image. The MEO can initiate the application by using the Virtualization Infrastructure Manager (VIM). After the application availability at the targeted host, a communication link is established to transfer the user context as the end user application is connected to the MEC application, the end user is not expected to be aware of the application mobility and the deployment of the application along with its state. A MEC stateful application needs to deliver the service continuity by importing the user context from the source MEH to the targeted MEH.

AdvantEDGE provides a support to the ETSI MEC API of the application mobility and allows the integration with the network scenario [252]. The API provides the support for relocation of user context between MEHs but the application instance relocation is not supported. The use case allows the MEC application user context transfer by using the API. The end-user devices are tracked and subscribed to the mobility procedure where the mobility manager receives the mobility notification of the end-user movement and the MEC application mobility. For running the application mobility experiment, the automation support is provided by AdvantEDGE for the User Equipment (UE) movement and the Point of Access (PoA) mobility.

8.3 Related Work

In [96], the authors discuss mobility-related issues, mainly focusing on the best instance to migrate the MEC application and what content to migrate to improve the Quality of Experience (QoE). Different mobility factors are taken into consideration. The work in [142] shows the optimal way to migrate the MEC application and the complete migration strategies to reduce energy expenditure. In [143], the authors consider the prototype system approach at the network layer to manage a seamless connection between the edge server and the mobile devices. Some of the works carried out the experimental tests using different MEC model, and migration strategies, one of them presents K8s as the MEO [144]. The work proposes reactive service migration with the evolved packet core (EPC). Other experimental studies present the integration of Open Source MANO (OSM), an orchestrator, with Open Network Edge Services Software (OpenNESS) [145], a MEC platform, to migrate the MEC applications between MEHs [146]. The study includes two components, one to maintain the application's state with the client and the other to focus on management.

In [147], the authors discuss various container and migration strategies focusing on the fog, edge, and cloud; the work focus on the current approaches and the framework for container-based services migration. In [148], the authors describe different methods of the migration of pod in K8s and present the results of downtime with and without migration along with the data size transferred. For stateful container migration, a prototype approach using an extended version of kubelet and customized containers is available on GitHub [149]. The prototype approach provides an extension of the *kubectl* command that includes a command for the checkpoint and migration of the running pod in K8s. This work presents the running pod migration across single or multiple clusters and adds the function necessary to the pod migration. Furthermore, the prototype implementation includes pod migration operator at control plane that has custom resource and the controller.

8.4 Emulated Network Scenario

AdvantEDGE platform is a MEEP that provides emulated and experimental environment with edge enabling technologies [236]. The platform runs on Docker and K8s, and provides experimentation with MEC deployment models along with their applications and services. AdvantEDGE supports some of the APIs and the edge services standardized by the ETSI MEC such as ETSI MEC 013 Location [25], ETSI MEC 012 Radio Network Information [23], ETSI MEC 028 WLAN Information [253], ETSI MEC 011 Edge Platform Application Enablement [27] and ETSI MEC 021 Application Mobility [81]. In addition to that, AdvantEDGE allows the changes of the location of devices within the network using their own APIs. The platform allows network characteristics configuration such as latency, jitter, throughput and packet loss that can be applied to the scenario. During the scenario deployment, containers run in the K8s pod. In each deployed pod, AdvantEDGE includes an companion container called as sidecar. The role of the sidecar is to apply the network characteristics from the simulation model. To implement simulation model, *TC-engine* is the responsible microservice, *tc* is called as Traffic Control. Whereas *tc-netem* technology is used for the network characteristics in each sidecar. AdvantEDGE supports different edge application and client deployment model. Furthermore, the platform allows mobility event, UE movement and mapping the geo-location of each elements. The UE movement can be monitored and visualized using the Geospatial Subsystem.

Fig. 8.1 shows the emulated network scenario based on AdvantEDGE platform. The scenario consists of one UE (ue1) in zone1, whereas zone2 and zone3 include an emulated MEHs edge1 and edge2 respectively along with PoAs. There are three different network access technologies representing different zones. The ue1 is able to connect to the MEHs via PoAs within each zone depending on the ue1 movement. The blue boxes are the MEHs. The brown boxs represent the mec-app for MEC application deployed on edge1 and vlc1 for ue1. The green box is the physical UE. The antennas represent the PoAs. Initially the MEC application runs on the MEH edge1 connected to a point on the network called Zone2. The edge1 MEC application and services are running externally to the platform. AdvantEDGE provides support to integrate the external MEC application and services within the scenario using an IP address and the port number. The zone elements represent a subnet, which can be composed by a set of network elements offering traffic transport service. Since the MEC architecture can be applied to any network technology, it is necessary to assume that further network elements are interposed between the infrastructures of an Internet Service Provider (ISP) and the edges of the network. These network elements are enclosed in logical zones and are not simulated by AdvantEDGE. In the Fig. 8.1, Operator1 is the ISP, which in the considered network scenario, provides the IP connectivity through the three access technologies and the IP services supported by means of the MEC architecture.

AdvantEDGE platform supports to allocate the physical locations of each elements presented in the scenario. The three different PoAs networking technologies are mapped in different geographical locations. Reference to the geographical scenario is in the Fig. 8.2. The scenario is an Arno River area in Pisa. The scenario considers three different access technologies: 4G, 5G and WiFi. The coverage radius of WiFi access technology is 200 meters (in red), whereas is 500 meters and 1000 meters for 5G (in orange) and 4G (in blue) respectively. In the figure, the blue line denotes the ue1 path considered for the experimentation.

The value of each metric of the determined optimum path between the MEH and the UE, as a function of the chosen PoA, are summarized in Table 8.1.

The GIS API (getGeoDataByName) was utilized during the tests to gather data on the geographical locations of the network's devices. These statistics make it possible to determine the client-to-PoA distance, which is critical to identify the set of PoAs that can provide connectivity to the

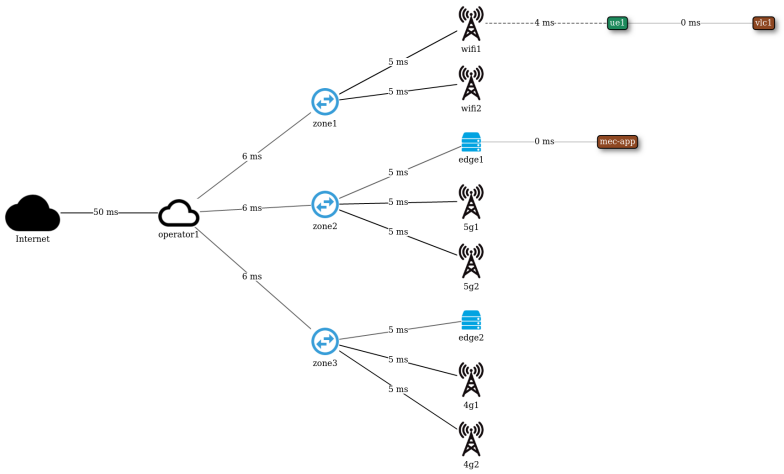


Figure 8.1: Network scenario described by the AdvantEDGE GUI

client. AdvantEDGE's Sandbox API (sendEvent) enables runtime PoA handover, which permits switching the PoA to which the client is connected.

8.5 Experimentation

The experimentation of the MEC application migration using the AdvantEDGE platform is divided into three phases. In the first phase, the deployment and the working structure are explained. The second phase deals the backend of the MEC application migration technique and integration with the AdvantEDGE platform. The third phase presents the experiment's migration flow, from the configuration deployment to the completion of the migration during the UE movement.

8.5.1 Description of the testbed

Fig. 8.3 shows the testbed overview with logical connectivity of the involved elements. The testbed is composed of 3 physical machines with specifications of GIGABYTE(32/512) Intel i7 NUCs. The NUC1 AdvantEDGE platform is deployed and runs the emulated network scenario implemented with

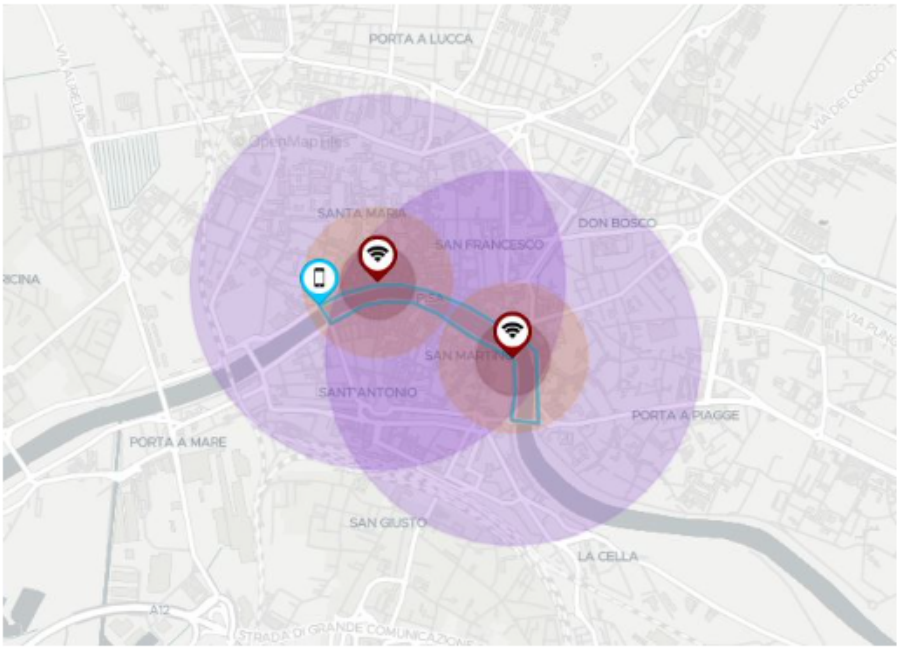


Figure 8.2: Map of the scenario considered in the experimental analysis with Advant-EDGE platform.

Table 8.1: Parameters of the path for the experimental tests

MEH-UE	WiFi-1	4G-1	5G-1	WiFI-2	4G-2	5G-2
PLoss (%)	0	0.016	0	0	0.0079	0
Jitter (ms)	12	16	13	16	15	17
Latency (ms)	26	31	30	26	39	30

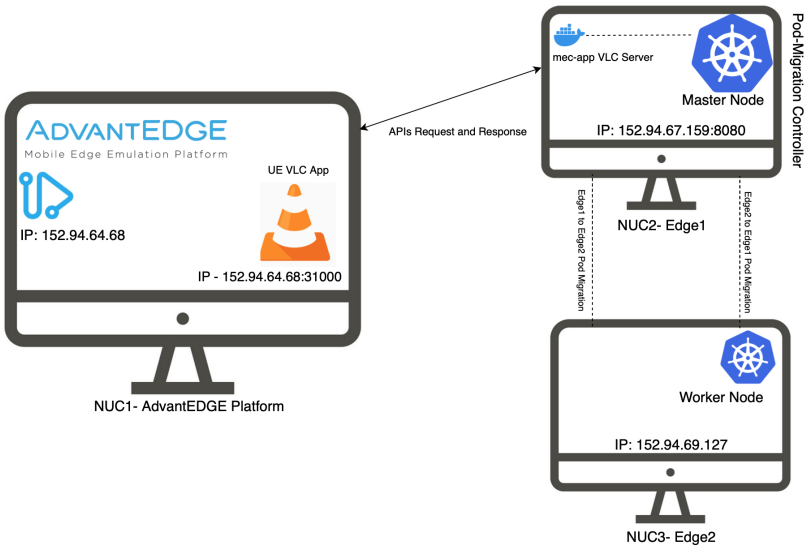


Figure 8.3: The testbed

AdvantEDGE and the UE application. As described in the Section 8.4, the emulated network scenario is composed by a set of APs, with three areas and two MEHs, edge1 and edge2. NUC2 enforces edge1 MEH where initially the mec-app is deployed and later migrated to the NUC3 edge2 MEH depending upon the UE movement. The AdvantEDGE platform is installed on a single K8s node on Ubuntu 20.04.4 LTS Operating System (OS). The AdvantEDGE platform GUI is accessed using the IP address 152.94.64.68, through which the emulated network scenario is configured and deployed. In the scenario configuration, the external mec-app is mapped with the edge1 mec-app using the IP address and the port number, called the external node integration. AdvantEDGE provides support for experimenting with external nodes and applications.

The management and deployment of the edge1 and edge2 MEHs are done using the cluster of K8s nodes: edge1 functions as the master node and edge2 as worker node. The MEHs interact with the AdvantEDGE platform using the API request and response provided by the AdvantEDGE platform. AdvantEDGE supports some of the ETSI MEC APIs. In particular, the location API, standardized by the ETSI GS as MEC 013 [25], is used to

track the information related to the UE physical location in the network during the experimentation. The mec-app is a video streaming application in a container deployed using K8s. The K8s run the application and allow access to the service using the IP address and port number. As the mec-app is deployed on the edge1 MEH, the video streaming is always accessed through the edge1 IP address and the specified port number, as shown in the Fig. 8.3. The UE application is a VLC application [239] running on the NUC1. The UE application reaches the mec-app video streaming service through AdvantEDGE. The mec-app service maps within the AdvantEDGE platform, where AdvantEDGE creates a mec-app and UE app pod using the external node integration.

8.5.2 Migrating MEC Application

The MEHs edge1 (master node) and edge2 (worker node) form a single cluster of nodes using K8s as depicted in Fig. 8.4. In the single cluster, edge1 deploys the mec-app, a video streaming pod. As initially in the emulated network scenario, the UE is connected to edge1; during UE movement mec-app is migrated to edge2. The single cluster node using the extended K8s version prototypical implementation is available on GitHub [149]. The prototype implementation includes components such as the K8s, containerd-cri with the extensions of CRIU, which is needed for runtime pod migration and podmigration-operator. The K8s insures the node synchronization within the cluster of nodes. The extended version of the K8s provides *kubectl-migrate* and *kubectl-checkpoint* commands [149]. In addition, edge1 is configured as the NFS server, whereas the worker node is the NFS client. The NFS shared folder results into giving access to the edge1 checkpoint storage where the mec-app is running. The pod migration API server directs the pod migration from edge1 to edge2 or vice-versa. The podmigration controller includes Customized Resource Definition (CRD) and a custom controller to watch the pod migration within the cluster of the nodes. CRD is a mechanism that supports user-defined data types in K8s and permits to design the required state while the controller can work towards the required state. The MEH edge1 runs the script to interact with the AdvantEDGE platform, where the UE information is exchanged using the APIs related to the AdvantEDGE platform and ETSI MEC specific.

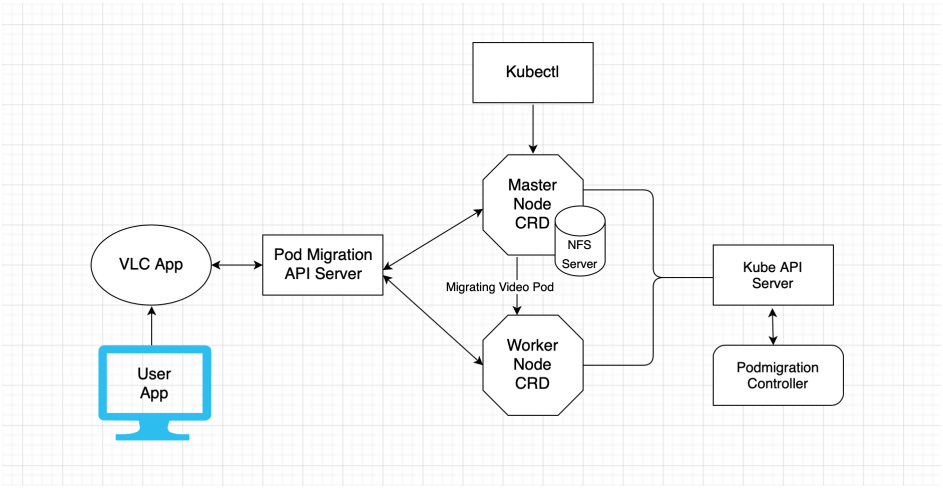


Figure 8.4: Video pod migration

8.5.3 Migration Flow

Fig. 8.5 presents the workflow of the MEC application migration between two MEHs using the AdvantEDGE platform. The emulated network scenario is created, configured, and deployed using the AdvantEDGE platform GUI. As soon as the scenario is deployed, AdvantEDGE creates a mec-app and UE app (referred to as vlc1) pod, which allows the UE app to reach the mec-app service via AdvantEDGE. The manager (script) registers the scenario information and the location of PoA and UE using the API. Initially, as configured in the scenario, vlc1 is closest and connected to the edge1 node. The vlc1 is connected, and the data is routed to the mec-app via edge1. The manager has pre-configured zone coverage for edge1 and edge2 depending upon the PoAs base station location. A PoA mobility event occurs during the UE movement, and the manager registers the UE location closer to the edge2. The manager orchestrates the scenario and guards the coordination with the podmigration-controller. The manager triggers and instructs the podmigration-controller for the mec-app migration from edge1 to edge2. The podmigration-controller starts the migration and checks if the source pod on edge1 is running or not. The podmigration-controller capture and contain the container state in a pod. Once source pod running information is acquired, a checkpoint of the source pod is created at the edge2 along with the checkpoint path. The edge2 confirms the checkpoint info creation; then, the pod is restored at edge2. The edge2

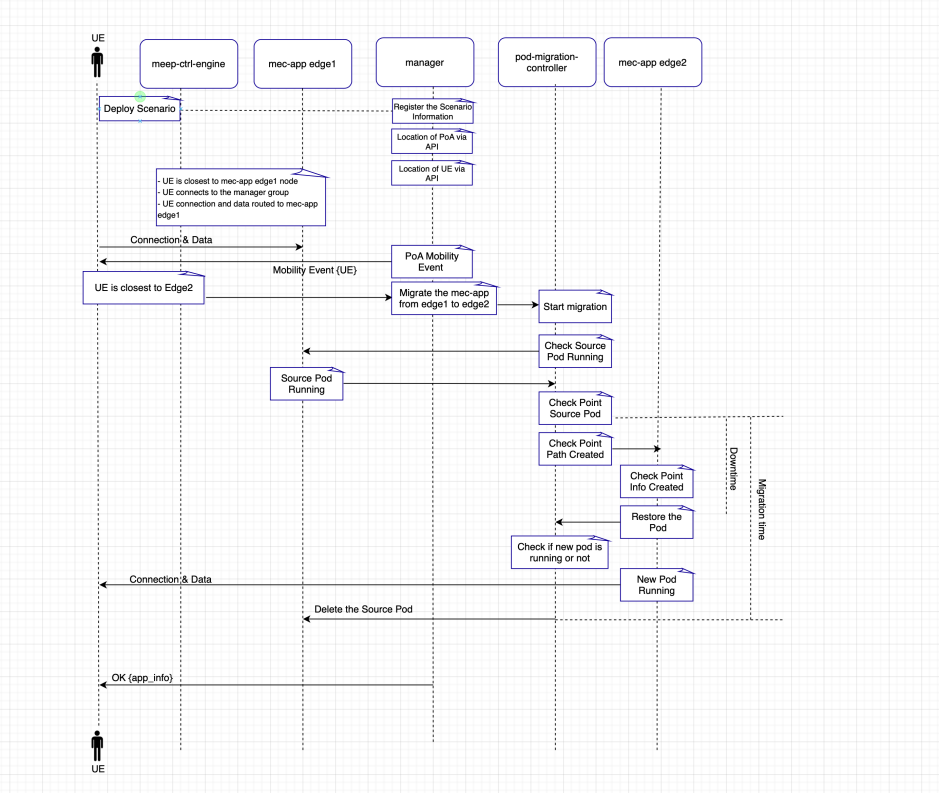


Figure 8.5: MEC application migration flow

informs podmigration-controller if the new pod is running or not. Once a new pod is running, vlc1 establishes the connection with the mec-app now running on edge2 via the manager. Later, the manager terminates the source pod with the help of the podmigration-controller. The manager affirms the app information with the UE application. In Fig. 8.5, the total migration time is noted from the start of the checkpoint to the completion of the migration process, whereas downtime is accounted for by resuming the pod on the destination MEH.

8.6 Experimental results

The test aims to determine the pod migration capabilities of the extended K8s and the different time-related performance parameters. Fig. 8.6 summarizes the different time-related parameters that can be measured during

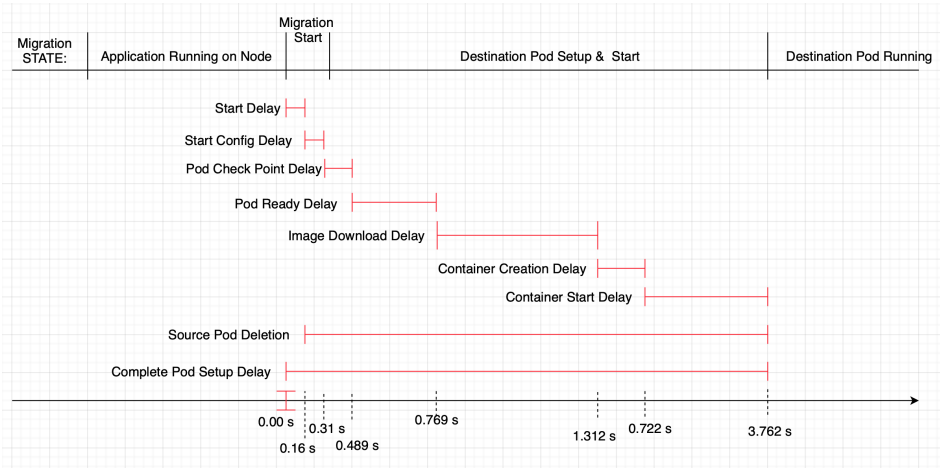


Figure 8.6: Migration process in edge K8s

the experimental sessions. Furthermore, the observed values during only one migration are reported in the bottom line. The shown results refer to the video streaming application migration with the extended migration techniques. The test reports that the average MEC application migration time value is 3.762 secs. The start of the delay is noted between the manager (script) to the pod migration operator. The start config delay mentioned in Fig. 8.6 relates to collecting information about running pods, such as whether the source pod is running or not and the current video state. The pod checkpoint delay is provided for establishing the checkpoint path with the destination host. The amount of time accumulated for the checkpoint pod's complete state is reported by the pod readiness delay. The destination host, who also produces the pod's container, gives the image download delay and container formation delay. The container start delay shows that the container is started and running for the end user. The pod migration operator deletes the source pod and records the total pod migration time in the logs. The extended K8s migration process indicates that this method allows a seamless migration process for video streaming applications. In comparison, the standard K8s is ill-advised for the seamless migration process of video streaming applications. Without the K8s plugin developed for migrating the application, the container migration via K8s adds more delay to this operation. At the same time, this problem could be avoided in the single cluster of nodes but is likely to affect the response time parameter in the multiple clusters within the Edge Datacenter.

Table 8.2: Description of time measured in the experiments.

$\Delta(ms)$	Total time taken for the application migration.
$\Delta_1(ms)$	Time taken by the manager to start the migration.
$\Delta_2(ms)$	Time accumulated for the Source Pod Checkpoint.
$\Delta_3(ms)$	Time for the Pod Ready Delay.
$\Delta_4(ms)$	Time to restore the Pod at destination.

A more deep analysis has been carried out observing a large set of migration events. In particular, these experimental tests are carried out between the two MEHs on the defined UE path as shown in Fig. 8.2. A total 100 number of migrations were taken place to observe the average migration time over the period. The extended version of the K8s and the NFS sharing helped to achieve a better and stable application migration latency. The MEC application migration time was noted from the pod migration operator logs with the help of the Kube API server. The time recorded in the logs depicts each stage of the pod migration from the creation of the checkpoint till the source pod's deletion. For the network layer, flannel is used. In K8s, the flannel supports the layer 3 networks between the multiple nodes across the single cluster, removing the port mapping complexities and providing the end user with a seamless migration experience.

Table 8.2 presents the set of time-related parameters that can be obtained from the K8s log, using the pod migration operator. The timeline dictate migration of the pod in the single cluster of nodes i.e. MEHs. Table 8.3 reports the statistical values of the Table 8.2 parameters during the experimental tests. The reported values refer to the statistics estimated observing 100 pod migration executions. Fig. 8.7 shows the mean and the 95% CI of the total migration time taken by the pod migration operator, as a function of the considered number of pod migration executions. The figure points out the large CI when only 10 migrations are observed. In this case the values are ranged from 2.7 secs to 4.5 secs. On the contrary, after the observation of 100 migrations, the 95% C.I. has a size of only 134.53 ms around the mean value.

Table 8.3: Time measurements (Averaging 100 independent migration executions)

	95% **C.I.**	**Median**	**Min**	**Max**	**95-th percentile**
$\Delta(ms)$	3015.46 ± 134.533	2866.5	1459	6789	3903.7
$\Delta_1(ms)$	19.4857 ± 1.1836	18.0	10	35	29.0
$\Delta_2(ms)$	8.1714 ± 0.6100	8.0	4	16	13.0
$\Delta_3(ms)$	7.971 ± 0.6087	7.0	14	16	12.549
$\Delta_4(ms)$	2995.371 ± 112.016	2860.0	2230	4471	3688.65

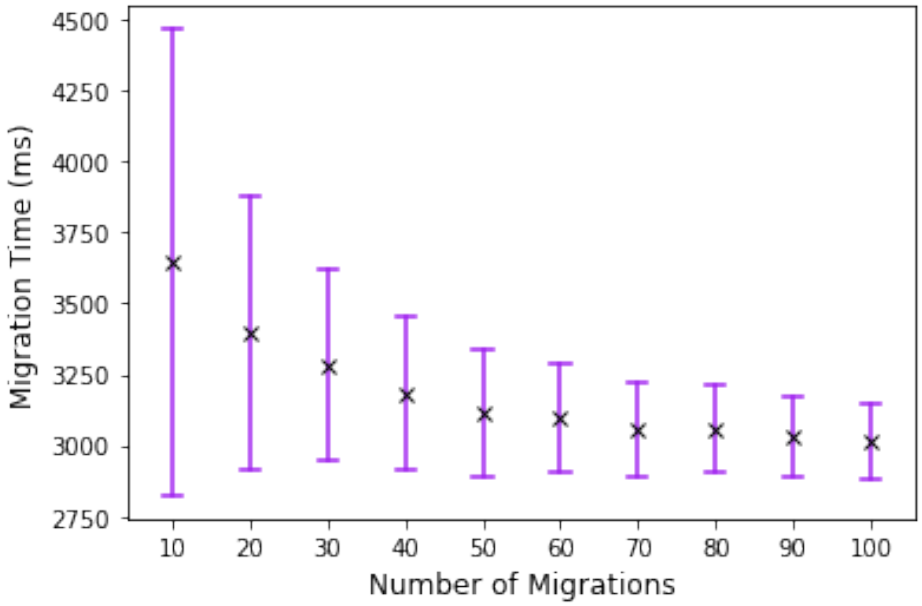


Figure 8.7: Confidence interval of MEC application migration

8.7 Conclusion

In this paper, we presented and studied a scenario to migrate the MEC applications by using K8s and AdvantEDGE. The experimental study presents an analysis of the MEC application migration by using the extended K8s pod migration strategies. We evaluated the pod migration strategies for the MEC applications and the time related to the pod migrations between MEHs.

Paper 4:
Joint multi-objective MEH selection and traffic path computation in 5G-MEC systems

Joint multi-objective MEH selection and traffic path computation in 5G-MEC systems

Prachi V. Wadatkar[1,2], Rosario G. Garroppo[2], Gianfranco Nencioni[1], Marco Volpi[2]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger

[2] Department of Information Engineering, University of Pisa

Published by the Elsevier Journal of Computer Networks, 2024

https://www.sciencedirect.com/science/article/pii/S1389128623006138


Computer Networks 240 (2024) 110168

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Joint multi-objective MEH selection and traffic path computation in 5G-MEC systems[☆]

Prachi Vinod Wadatkar[a,b], Rosario Giuseppe Garroppo[b], Gianfranco Nencioni[a,*], Marco Volpi[b]

[a] University of Stavanger, Stavanger, 4021, Norway
[b] University of Pisa, Pisa, 56126, Italy

ARTICLE INFO

Keywords:
5G
Multi-access Edge Computing
Multi-objective Dijkstra Algorithm
Pareto-optimal
MEH selection
Traffic path computation
AdvantEDGE

ABSTRACT

Multi-access Edge Computing (MEC) is an emerging technology that allows to reduce the service latency and traffic congestion and to enable cloud offloading and context awareness. MEC consists in deploying computing devices, called MEC Hosts (MEHs), close to the user. Given the mobility of the user, several problems rise. The first problem is to select a MEH to run the service requested by the user. Another problem is to select the path to steer the traffic from the user to the selected MEH. The paper jointly addresses these two problems. First, the paper proposes a procedure to create a graph that is able to capture both network-layer and application-layer performance. Then, the proposed graph is used to apply the Multi-objective Dijkstra Algorithm (MDA), a technique used for multi-objective optimization problems, in order to find solutions to the addressed problems by simultaneously considering different performance metrics and constraints. To evaluate the performance of MDA, the paper implements a testbed based on AdvantEDGE and Kubernetes to migrate a VideoLAN application between two MEHs. A controller has been realized to integrate MDA with the 5G-MEC system in the testbed. The results show that MDA is able to perform the migration with a limited impact on the network performance and user experience. The lack of migration would instead lead to a severe reduction of the user experience.

1. Introduction

Some of the use cases defined for 5G-and-beyond systems [1,2] require high performance, in terms of latency, reliability, throughput, etc. [3]. Multi-access Edge Computing (MEC) is one of the fundamental technologies committed to satisfying these requirements. MEC consists in the computing platforms located near the users. MEC deployments in proximity to users often demand additional computing resources and infrastructure to ensure optimal performance. This additional effort is largely compensated by the achieved performance (not only low latency, but also reduction of network congestion and increased data rate) and the enabling of new advanced services (which can also rely of context awareness). As described in [4], the main functions of 5G and MEC architecture interact as shown in Fig. 1.

In particular, the user requests an application to the Application Function (AF). The MEC may be seen as an AF, and if it is trusted it may directly interact with the 5G Network Functions (NFs). In the case of untrusted MEC (as a general untrusted AF), the interaction MEC-5G system is mediated by the Network Exposure Function (NEF) that is in charge of securely exposing the network capabilities and events to untrusted AFs. A key element of the MEC system level is the MEC Orchestrator (MEO), which provides centralized functions for orchestrating the computing resources and operation of the MEC hosts (MEHs). On one side, the MEO interacts with the different MEHs to acquire information on the available resources and Apps. On the other side, MEO interacts with the NEF (or directly with the 5G NFs), to acquire capability and state information about the 5G Core Network (CN) services. The MEO uses these data to select the MEH for instantiating a new application request (or migrating a running application) by considering the performance requirements (e.g., latency, throughput, packet loss, etc.), the MEH available resources, and the MEH and network performance.

During the service, through the NEF (or directly interrogating the 5G NFs in case of trusted MEC), the MEO acquires UE-related events of interest to decide if some actions (e.g. change the access point, modify the traffic path) on the 5G domain needs to be performed to satisfy the required QoS. On the MEC infrastructure, the MEO monitors the performance given by the MEH. Problems in the network domain or in

☆ This work was supported by the Norwegian Research Council through the 5G-MODaNeI project (no. 308909), and by the Italian Ministry of Education and Research (MIUR) in the framework of the FoReLab project (Departments of Excellence).
* Corresponding author.
E-mail address: gianfranco.nencioni@uis.no (G. Nencioni).

https://doi.org/10.1016/j.comnet.2023.110168
Received 10 July 2023; Received in revised form 18 November 2023; Accepted 27 December 2023
Available online 3 January 2024

Abstract:
Multi-access Edge Computing (MEC) is an emerging technology that allows to reduce the service latency and traffic congestion and to enable cloud offloading and context awareness. MEC consists in deploying computing devices, called MEC Hosts (MEHs), close to the user. Given the mobility of the user, several problems rise. The first problem is to select a MEH to run the service requested by the user. Another problem is to select the path to steer the traffic from the user to the selected MEH. The paper jointly addresses these two problems. First, the paper proposes a procedure to create a graph that is able to capture both network-layer and application-layer performance. Then, the proposed graph is used to apply the Multi-objective Dijkstra Algorithm (MDA), a technique used for multi-objective optimization problems, in order to find solutions to the addressed problems by simultaneously considering different performance metrics and constraints. To evaluate the performance of MDA, the paper implements a testbed based on AdvantEDGE and Kubernetes to migrate a VideoLAN application between two MEHs. A controller has been realized to integrate MDA with the 5G-MEC system in the testbed. The results show that MDA is able to perform the migration with a limited impact on the network performance and user experience. The lack of migration would instead lead to a severe reduction of the user experience.

9.1 Introduction

Some of the use cases defined for 5G-and-beyond systems [254, 255] require high performance, in terms of latency, reliability, throughput, etc. [256]. Multi-access Edge Computing (MEC) is one of the fundamental technologies committed to satisfying these requirements.

MEC consists in the computing platforms located near the users. MEC deployments in proximity to users often demand additional computing resources and infrastructure to ensure optimal performance. This additional effort is largely compensated by the achieved performance (not only low latency, but also reduction of network congestion and increased data rate) and the enabling of new advanced services (which can also rely of context awareness).

As described in [18], the main functions of 5G and MEC architecture interact as shown in Figure 9.1.

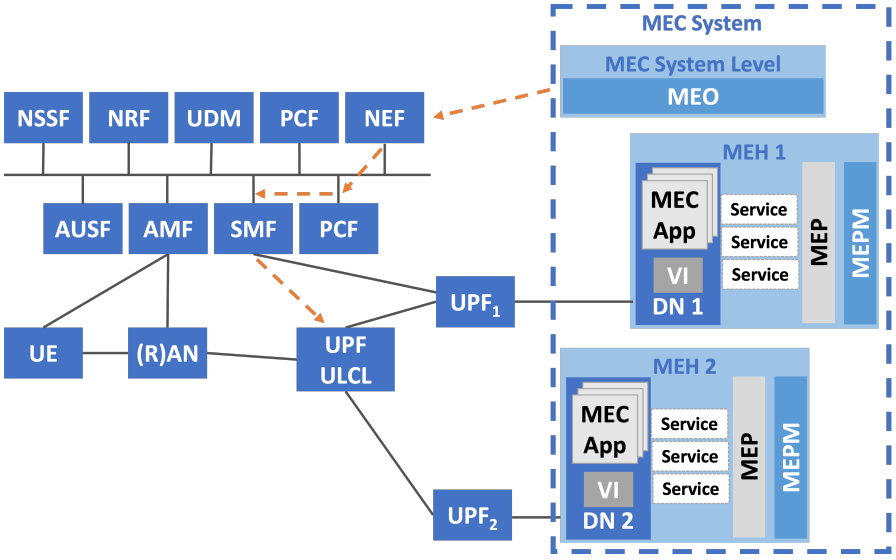


Figure 9.1: Interaction 5G-MEC functions for supporting services.

In particular, the user requests an application to the Application Function (AF). The MEC may be seen as an AF, and if it is trusted it may directly interact with the 5G Network Functions (NFs). In the case of untrusted MEC (as a general untrusted AF), the interaction MEC-5G system is mediated by the Network Exposure Function (NEF) that is in charge of

securely exposing the network capabilities and events to untrusted AFs. A key element of the MEC system level is the MEC Orchestrator (MEO), which provides centralized functions for orchestrating the computing resources and operation of the MEC hosts (MEHs). On one side, the MEO interacts with the different MEHs to acquire information on the available resources and Apps. On the other side, MEO interacts with the NEF (or directly with the 5G NFs), to acquire capability and state information about the 5G Core Network (CN) services. The MEO uses these data to select the MEH for instantiating a new application request (or migrating a running application) by considering the performance requirements (e.g., latency, throughput, packet loss, etc.), the MEH available resources, and the MEH and network performance.

During the service, through the NEF (or directly interrogating the 5G NFs in case of trusted MEC), the MEO acquires UE-related events of interest to decide if some actions (e.g. change the access point, modify the traffic path) on the 5G domain needs to be performed to satisfy the required QoS. On the MEC infrastructure, the MEO monitors the performance given by the MEH. Problems in the network domain or in the MEC resources can lead the MEO to modify the traffic steering rules and/or migrate the service to another MEH.

Intuitively, to maintain high performance, a user request should be served by the nearest MEH that runs the required application. Empirical studies confirm this intuition [257]. Maintaining application proximity for mobile users poses significant challenges. Among these, two fundamental challenges regarding the achievement of a desirable trade-off between QoS and migration cost are related to the questions: 1) How are the applications migrated [258][259]? and 2) When/where are the applications migrated [260]? This paper considers this second question by analyzing two important problems: the dynamic selection of the MEH providing the resources for running the application requested by a UE (*MEH selection*) and the computation of the traffic path between the selected MEH and the UE (*traffic path computation*). Both problems aim to maintain the requested QoS as the UE moves. Indeed, a major challenge in 5G MEC is related to UE mobility. To ensure the desired QoS parameters, including low latency and minimal packet loss, it may become necessary to modify the traffic path and, in certain situations, switch the MEH, leading to what is known as an MEH handover. This process can be due to the deterioration of performance provided by the MEH and/or the traffic path leading to the source MEH.

In the latter scenario, performance degradation may arise from handovers between different base stations or an increase in traffic load along this path.

9.1.1 Problem Definition and Contribution

The paper jointly considers the *MEH selection* problem and the *traffic path computation* problem in 5G-MEC systems by defining a multi-objective optimization problem. The heterogeneity of the 5G-MEC applications leads to defining new control algorithms aimed to consider different criteria simultaneously. For example, some applications require the minimization of the latency, while others require minimizing packet loss, increasing the security of the data transfer and of the elements supporting the application, minimizing the energy consumption, etc.. Hence, to account for most requirements the optimization problems cannot consider a single criterion. The complexity of the multi-objective optimization problem is obviously higher than the single criterion one. For example, for the routing problem, algorithms, such as Dijkstra, allow the computation of the single criterion minimum cost path in an efficient way. In contrast, different studies have been focused on finding efficient approaches for solving the complex multi-objective counterpart [261]. The most common approach is to define an optimization function which is a weighted combination of all criteria. In this manner, the multi-criteria problem can be treated as a single-criterion problem, i.e. with reduced complexity. However, this approach implies that the weights need to be established a priori. Before the problem solution begins, the relative "importance" of the different criteria needs to be set. The solution is optimal only for the selected weighting. The heterogeneity of the applications of today networks does not allow finding a weighting providing good results for all.

The multi-criteria optimization problem gives a set of Pareto optimal solutions (the so-called Pareto front), which can be dynamically selected depending on the requirements of the particular application. The MEO determines the most suited solution for the application needs, analyzing the Pareto front. Then, MEO decides where to locate (or re-locate) the application and how to steer the traffic. Although multi-criteria optimization is NP-hard, recent works proposed efficient algorithms for typical-size networks that work well in practice.

The main contribution can be summarized in the following points.

- To jointly consider the problems of MEH selection and traffic path computation, the paper proposes a procedure for defining a graph able to account for both the network-layer and application-layer performance. In this manner, the two problems are mapped in the single problem of finding optimized paths between the user and the MEHs.
- To solve the problems using the defined graph, the paper proposes the Multi-objective Dijkstra Algorithm (MDA) for computing the Pareto front of the Multi-Objective Shortest Path (MOSP) model. Furthermore, the paper shows how the Pareto front is used for supporting applications with different requirements.
- To evaluate the performance of MDA, the paper implements a hybrid testbed. The testbed includes simulative, emulative and experimental parts.
- To integrate the MDA with the 5G-MEC system in the testbed, the paper realizes a controller to retrieve the input of MDA and to apply the output of MDA for both network layer and application layer.

The paper is organized as follows. Section 9.2 summarizes the MOSP problem and the MDA algorithm, while Section 9.3 describes the proposed approach for jointly solving the MEH selection problem and the traffic path computation problem, using the MDA algorithm. Sections 9.4 and 9.5 introduce extensive evaluations in an emulative/experimental environment. Section 9.6 presents the related works and summarizes the novelties of this paper. Finally, Section 9.7 concludes the paper.

9.2 MOSP Problem and MDA Algorithm

This section summarizes the key definitions and features of the MOSP problem and of the MDA algorithm.

9.2.1 MOSP Problem

The technical details on the MOSP problem definition and on the MDA algorithm can be found in [262]. The definitions of the different symbols used in this summary are shown in Table 9.1.

Table 9.1: List of used symbols.

Symbols	**Definition**
$G = (V, A)$	Directed graph without parallel arcs, composed by the set V of nodes and the set A of arcs
$N = \|V\|$	Number of nodes
$M = \|A\|$	Number of arcs
d	Number of attributes (e.g. delay, energy consumption, jitter, packet loss, etc.) considered by the MOSP
c_a	Vector of values assumed by the attributes in the arc a, $c_a(c_{a1}, c_{a2}, ..., c_{ad})$
$\delta^+(v)$	Set of outgoing arcs in v
$\delta^-(v)$	Set of incoming arcs in v
(v_1, v_k)**-path,** $P_{(v_1,v_k)}$	Set of arcs necessary to reach node $v_k \in G$ from node $v_1 \in G$. It can be represented by the sequence of traversed nodes, i.e. $P_{(v_1,v_k)} = (v_1, v_2, ..., v_k)$
$\mathbf{c}_{(v_i,v_{i+1})}$	Cost vector related to the arc $(v_i, v_{i+1}) \in G$
$\mathbf{c}(P_{(v_1,v_k)})$	Cost of the (v_1, v_k)-path. In the case of summable attributes, it can be calculated as $\mathbf{c}(P_{(v_1,v_k)}) = \sum_{i=1}^{k-1} \mathbf{c}_{(v_i,v_{i+1})}$
$P <_D Q$	Path P dominates path Q

The solution of the MOSP problem requires the definition of the concept dominated-path, $P_{(v_1,v_k)}$.

Let us consider two paths $P_{(s,t)}$ and $Q_{(s,t)}$ with the same origin node s and the same destination node t. Let $c_j(P)$ and $c_j(Q)$ the j-th component of the cost vector of the two paths. Then $P <_D Q$

$$\begin{aligned} If \quad & c_j(P) \leq c_j(Q), \quad \forall j \in [1, d] \\ & \exists i \in [1, d] \quad : \quad c_i(P) < c_i(Q) \end{aligned} \tag{9.1}$$

It is worth noting that (9.1) considers only the cost vectors of the two paths. Hence, in general given two cost vectors, α and β, (9.1) establishes the dominance between two cost vectors, i.e. if $\alpha <_D \beta$.

Given a (s,t)-path P with cost vector $\mathbf{c}(P)$, $\mathbf{c}(P)$ is called a *non-dominated vector* if there is no other (s,t)-path having a cost vector dominating $\mathbf{c}(P)$. In this case, P is denoted as *efficient path*. There is no (s,t)-path in G that dominates P.

The MOSP problem aims at finding a path optimizing simultaneously different attributes, which often are conflicting. An improvement in one objective can lead to the worsening of at least one other objective. In this scenario, the solution of the MOSP problem is represented by the minimum complete set of efficient paths between a node s and every node $v \in V$. For each node pair (s,v), this set contains exactly one efficient path per non-dominated cost vector. This definition refers to the so-called one-to-all variant. If a target node t is given as input, the one-to-one variant of MOSP aims at finding the minimum complete set of efficient paths connecting the nodes s and t.

9.2.2 MDA Algorithm

The MDA is a recent label-setting algorithm [262] for solving the MOSP problem. A more recent paper presents further improvements aimed at reducing the solving time of the one-to-one version of MOSP [263]. The performance comparison among various one-to-many MOSP algorithms demonstrates the superiority of MDA, particularly on graphs comprising approximately 300,000 nodes and 800,000 links [264]. Notably, these graph sizes are often larger than real-world networks. Like other label-setting algorithms, the MOSP uniquely represents a path P as a sequence of node labels. For each node v, a label is composed by three values $l_v = (v, \mathbf{c}_{l_v}, l_{pred})$, where:

- v is the identity of the node owner of the label
- $\mathbf{c}_{l_v}$ is the cost vector of the (s,v)-path
- l_{pred} is the pointer to a label of a predecessor node $u \in \delta^-(v)$. This label refers to the (s,u)-subpath of the (s,v)-path.

The defined labels can be compared to each other. The labels comparison and the one-to-one correspondence between paths and labels allow finding the efficient paths analyzing the labels. In other words, the efficient paths can be found through the list of non-dominated labels of the nodes. The definition of non-dominated labels is as follows.

Given two labels, $l_1(v)$ and $l_2(v)$ corresponding to two alternative (s, v)-paths P_1 and P_2

- $l_1(v)$ dominates $l_2(v)$, indicated as $l_1(v) \leq_D l_2(v)$, and both are non-equivalent if and only if $\mathbf{c}_{l_1(v)} <_D \mathbf{c}_{l_2(v)}$ and $\mathbf{c}_{l_1(v)} \neq \mathbf{c}_{l_2(v)}$

For each node v, the MOSP commonly leads to having a set of labels, indicated as L_v, that are non-dominated. To compare a new label $l_{new}(v)$ with L_v the following definition is necessary.

- L_v dominates $l_{new}(v)$ or $L_v \leq_D l_{new}(v)$ iff there is a label $l_\alpha \in L_v$ s.t. $l_\alpha \leq_D l_{new}(v)$.

The lexicographic order of labels is defined as follows. Let $l_1(v)$ and $l_2(v)$ be two labels corresponding to two alternative (s, v)-paths. The label $l_1(v)$ is lexicographically smaller than $l_2(v)$, denoted as $l_1(v) <_{lex} l_2(v)$, iff $\mathbf{c}_{l_1(v)}$ is lexicographically smaller than $\mathbf{c}_{l_2(v)}$, and this is true if $\mathbf{c}_{l_{1,k}(v)} < \mathbf{c}_{l_{2,k}(v)}$ for the first index $k \in \{1, \ldots, d\}$ such that $\mathbf{c}_{l_{1,k}(v)} \neq \mathbf{c}_{l_{2,k}(v)}$.

Algorithm Description

In the following, a brief description of the MDA algorithm is presented. Further details can be found in [262]. The algorithm defines the following set of lists and vectors:

- H: List storing the tentative labels in lexicographical order. The tentative labels correspond to paths explored during the algorithm but for which it is not yet decided if they are non-dominated paths. At any point during the algorithm, H stores at most one label per node, i.e. the size of H is bounded N.

- L_v: For each node $v \in V$, a list L_v contains the non-dominated labels.

- **lastProcessedLabel**: This vector contains the pointer to the last processed label for each arc in the graph.

Algorithm 1 Pseudocode of MDA

Input: Graph $G = (V, A)$ with cost vectors $\mathbf{c}_(v_i, v_(i+1)) \in \mathbb{R}^d$, Node $s \in V$
Output: set L_v of non-dominated labels $\forall v \in V$
Priority Queue $H \leftarrow 0$
for $v \in V$ **do**
  Efficient Labels $L_v \leftarrow 0$
end for
$L \leftarrow \bigcup_{v \in V} L_v$
for $a \in A$ **do**
  `lastProcessedLabel`$[a] \leftarrow 0$
end for
Label $l_s \leftarrow [s, (0, \cdots, 0)$ `NULL`$]$
$H \leftarrow H.insert(l_s)$
while $H \neq 0$ **do**
  $l_v^* \leftarrow H.extract\_lexmin()$
  $v \leftarrow l_v^*.node$
  $L_v.push\_back(l_v^*)$
  $l_v^{new} \leftarrow$ `nextCandidateLabel`$(v$,`lastProcessedLabel`$,\delta^-(v), L)$
  if $l_v^{new} \neq$ `NULL` **then**
    $H.insert(l_v^{new})$
  end if
  for $w \in \delta^+(v)$ **do**
    $H \leftarrow$ `propagate`(l_v^*, w, H, L)
  end for
end while
return L_v for all $v \in V$

Algorithm 1 shows the pseudocode of MDA. The initialization phase sets up the empty list H and the N empty lists L_v. The L_v lists are grouped into a list L. Then, there is the initialization of the vector $lastProcessedLabel$ composed of M values equal to zero. Then the algorithm generates a label for the origin node s which is $l_s = (s, (0, \ldots, 0), NULL)$ and inserts them in the list H.

After this phase, the `while` loop begins and lasts until the list H becomes empty. An iteration of the `while` loop starts with the extraction of the lexicographical smallest label l_v^* from the list H, where v is the node associated with the label. It is worth recalling that in H the labels are lexicographically sorted, then the label l_v^* is surely non-dominated. Thus the label can be added to the end of the list of non-dominated labels of the node v, i.e. L_v. This is the only way a label is added to the list L_v, i.e. the label gets permanent. As a consequence, the sets L_v $v \in V$ are also lexicographically sorted. Each iteration carries out two main tasks. The first is to find the next *tentative/candidate* label for node v that can be added to H. This task is necessary because only one label per node v is present in H. Hence, when this label is extracted, a new tentative label for v must be found (if it exists) and added to H. The search of the new label, l_v^{new}, is performed extending existing non-dominated labels at the predecessor node $u \in \delta^-(v)$ along the arc (u,v). In particular, l_v^{new} must be lexicographically the smallest and the non-dominated one among the all possible extension, i.e.

$$l_v^{new} = arglexmin_{l,u}\{lv = (v, c_l + c_{uv}, l) | Lv \not\prec_D l_v\}$$

where $l \in L_u$ and $u \in \delta^-(v)$.

This part allows maintaining a single tentative label for each node, which represents an important characteristic that differentiates the MDA from the classical label-setting MOSP algorithms, which keep a set of tentative labels for the same node.

The second task is to propagate the extracted l_v^* to the successor nodes $w \in \delta^-(v)$. Let $l_w = (w, c_{l_v^*} + c_{vw}, l_v^*)$ be such a propagated tentative label. If l_w is dominated by any label in L_w, it is discarded. Otherwise, if there is no label for w in H, l_w is inserted. On the contrary if there is a label, a comparison between l_w and the label of w already present in H is performed. The lexicographically smaller will be in H, the other will be discarded. The details and the pseudocode of these two tasks can be found in [262].

The running time performance of MDA is deeply analyzed in [262] with large synthetic and real-world graph instances. In the one-to-all case, the results show that for $d \geqslant 3$ the running time is $O(d \cdot (\log(N) \cdot \sum_{v \in V} |L_v| + M \cdot (\max_{v \in V} L_v|)^2)$. To the best of the authors' knowledge, MDA is the fastest among any MOSP algorithm known so far.

9.3 Proposed Multi-layer Graph

To jointly address the MEH selection problem and the traffic path computation problem, the approach of incorporating information on the MEH performance at the application layer in a graph is proposed. In general, the two problems are considered separately. As an example, the graph considering only the network-layer information is built taking into account the network topology. Routing algorithms, such as MDA, can be used to compute the traffic path from the user to the chosen MEH supporting the requested application. Before starting the traffic path computation, the MEH needs to be selected. The MEH selection can be done using algorithms such as [265]. The problem with this disjoint approach is that the algorithm used for the MEH selection considers only aspects related to the MEH platform, such as the constraints on the computation and communication capacity. The goal of the algorithm is to maximize the number of served requests while assuring that the service placement cost is within the budget. The quality of the service experienced by the application user is not taken into account. Indeed, bad quality is experienced if there is no traffic path between UE and selected MEH able to guarantee the network performance required by the application.

The proposed approach is based on the idea of enhancing the network graph by incorporating MEH performance information. For each MEH, a node and an arc are introduced to the the graph. For each MEH, a node and an arc are introduced to the network graph. The added node represents the application layer of the MEH, while the arc captures the application layer performance of the MEH in processing data exchanged with the network layer. The attributes of this arc reflect the MEH performance from the perspective of the running application and can be obtained through monitoring tools implemented within the MEH.

The remaining part of the graph considers the available alternative network paths between the client and each MEH supporting the required application. The values of the attributes of each arc of the graph are

estimated on each link of the network. In some cases, such as on the wireless network interface, the metrics are estimated from the values reported in some APIs defined by the MEC architecture.

Figure 9.2 shows an example of the graph obtained when the user (node `UE`) asks for a service that can be offered by four different MEHs located in nodes `5`, `6`, `7` and `8`. In each of these nodes, the MEH is represented by the extra node in red (i.e. `MEH5A`, `MEH6A`, `MEH7A` and `MEH8A`). The added extra arcs representing the MEH performance at the application layer are in red. The UE has three alternative Points of Access (PoAs) for accessing the network: `PoA1`, `PoA2` and `PoA3`. The black arcs of the graph refer to the network layer. The vector of each arc gives the values of the considered attributes. In the example, the selected attributes are respectively the packet loss probability, the jitter and the latency. However, other attributes can be used, such as security, energy consumption, etc.

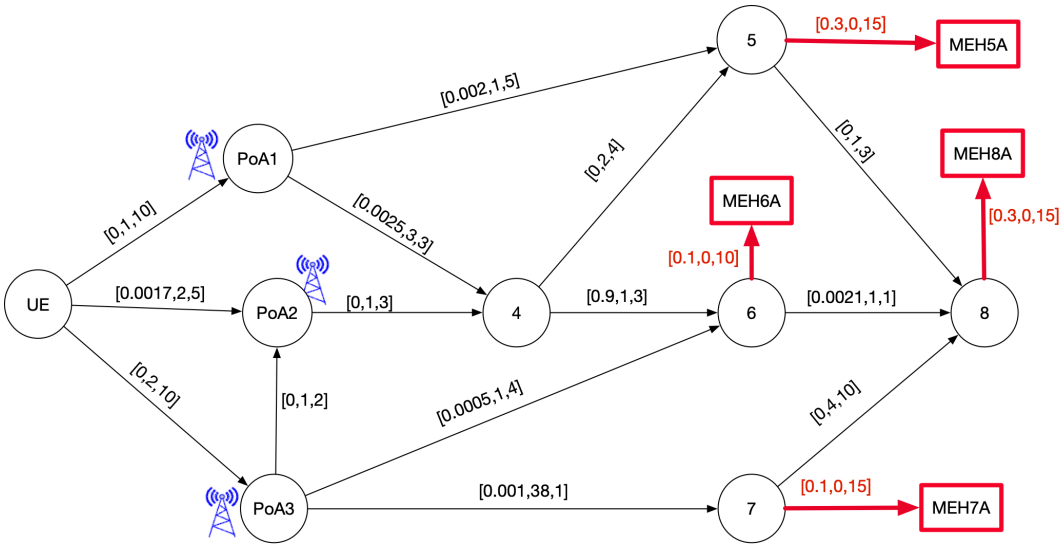


Figure 9.2: An example of an extended graph. The nodes labeled as `MEH5A`, `MEH6A`, `MEH7A`, and `MEH8A` represent different MEHs, while the nodes `PoA1`, `PoA2`, and `PoA3` represent the available PoAs for the user `UE`. Node `4` is exclusively utilized for traffic routing, while the remaining nodes, apart from traffic routing, fulfill the functions of the MEH. Each link is associated with a cost vector denoting packet loss probability, jitter, and latency. The black arcs represent network-layer links, while the red arcs indicate internal links connecting the application layer of the MEH with the network layer.

This graph serves as the input to the MDA, which generates the Pareto front of paths from the client to each candidate MEH, considering the

application-level performance. The output of the MDA is utilized to select the MEH and determine the corresponding path.

9.3.1 Metrics

There are different algorithms for finding the entire set of Pareto-optimal paths in the graph presented above. These algorithms, such as MDA, are designed for sum and bottleneck-type metrics. In the case of bottleneck-type metrics, a simple strategy is to prune from the graph the arcs that do not satisfy the constraints on the bottleneck metric. For example, if the service requires minimum datarate, the simple "pruning" of the arcs non-satisfying this constraint can be applied to the original graph before running the algorithm. Furthermore, information on the maximum datarate given by a particular path can be obtained considering that the maximum datarate offered by a path corresponds to the lowest datarate observed in its arcs. In [266], the authors suggest an alternative approach to deal with a bottleneck-type metric: they suggest converting it into a sum-type by using reciprocals. For example, in the case of the available datarate, this bottleneck-type metric can be converted into a sum-type defining the optimal goal as $f^p = \sum_{a \in A_p}(1/datarate(a)$, where p is a path and A_p is the set of arcs of p.

Referring to the presented example, it is important to note that latency is a sum-type metric, while the other two metrics, namely packet loss and jitter, require additional assumptions and manipulation.

Regarding packet loss, it is necessary to assume the independence of the loss process across consecutive arcs and, more generally, across diverse arcs. The packet loss probability P_{loss} on a path L, composed by two consecutive arcs (s, u) and (u, v) with independent packets losses, can be computed by means of the probability of the complementary event "correct packet delivery" (CPD) , i.e. $P_{loss} = (1 - P_{CPD})$. The independence of the packet loss on the links (s, u) and (u, v) composing the path L leads that the CPD probability on L is given by the product of the CPD probability on both links of the path, i.e. $P_{CPD_L} = P_{CPD_{s,u}} P_{CPD_{u,v}}$. Hence, P_{loss} of the path L can be calculated with the relation $P_{loss_L} = 1 - P_{CPD_{s,u}} P_{CPD_{u,v}}$.

This result can be easily generalized to a generic (s, d)-path P composed by k arcs:

$$P_{loss_P} = 1 - \prod_{i=1}^{k-1} P_{CPD_i}. \tag{9.2}$$

The minimization of P_{loss_P} can be found maximizing $\prod_{i=1}^{k-1} P_{CPD_i}$ or, in other words, minimizing $(-\prod_{i=1}^{k-1} P_{CPD_i})$. Using the logarithms property: $-\log(\prod_{i=1}^{k-1} P_{CPD_i}) = -\sum_{i=1}^{k} \log_{10}(P_{CPD_i})$. Hence, in the case of the packet loss attribute, the sum-type metric for the arc i is represented by $-\log_{10} P_{CPD_i}$. Using this metric for each link of the graph, the MDA will find the path with the minimum value of $-\sum_{i=1}^{k} \log_{10}(P_{CPD_i})$, which corresponds to the path with the minimum P_{loss}.

Two important assumptions are required when considering jitter. Firstly, it is assumed that jitter is defined as the variance of latency. Secondly, the latency in each link is modeled as a Gaussian random process, and the processes associated with different links are assumed to be independent. These assumptions enable the estimation of jitter as the variance of the Gaussian model of latency in each link. Furthermore, by assuming the independence of the processes modeling latency in diverse arcs, the jitter can be transformed into a sum-type metric. Indeed, these assumptions lead to calculating the jitter of a path L, composed by two consecutive independent arcs (s, u) and (u, v), by means of the sum of the jitter of each arc:

$$jitter_L = jitter_{s,u} + jitter_{u,v}. \tag{9.3}$$

9.3.2 Graph without MEH State Information

This case refers to the disjoint approach, wherein the selection of the MEH is determined using dedicated algorithms, while the MDA is solely utilized for defining the traffic path between the user and the chosen MEH. Referring to the example of Figure 9.2, MDA receives as input the subgraph $G = (V, A)$ derived from the graph in the figure after removing all the red arcs and red nodes, which correspond to the application layer information..

Let us assume that the dedicated algorithm has selected the MEH in node `8`. In this scenario, the MDA will output the set of all non-dominated paths between the `UE` and `8`. In general, it provides all non-dominated paths between `UE` and any node of the graph. In each node, the labels related to the alternative non-dominated paths are lexicographically ordered. The output is presented in Table 9.2: each row represents the list of non-dominated labels between the `UE` and each node. For sake of clarity, each label includes the node owner, the cost vector, and the link to the previous node. The link is represented as a pair, where the first element is the identifier of the previous node and the second is the identifier of the label in the previous node. This information is necessary because each node can

have multiple non-dominated labels, and each label may correspond to a different path. It is worth noting that in the destination node `8`, the MDA provides 6 distinct non-dominated labels. Therefore, selecting one path from the non-dominated set requires defining a strategy that considers the application requirements. For example, when the strategy aims to minimize P_{loss}, the path `UE-PoA3-7-8` is selected. On the other hand, if the goal is to achieve minimum latency with $P_{loss} \leq 0.01$, the available paths are `UE-PoA3-4-5-8` (with lower P_{loss}) and `UE-PoA3-6-8` (with lower jitter).

Table 9.2: Output of the MDA algorithm for the example "Graph without MEH State Information".

Node	Non-dominated Labels
UE	[**UE**, [0, 0, 0], ['NULL', 'NULL']]
PoA1	[**UE**, [0.0, 1, 10], [**UE**, 1]]
PoA2	[**PoA2**, [0.0, 3, 12], [**PoA3**, 1]], [**PoA2**, [0.0017, 2, 5], [**UE**, 1]]
PoA3	[**PoA3**, [0.0, 2, 10], [**UE**, 1]]
4	[**4**, [0.0, 4, 15], [**PoA2**, 1]], [**4**, [0.0017, 3, 8], [**PoA2**, 2]]]
5	[**5**, [0.0, 6, 19], [**4**, 1]], [**5**, [0.0017, 5, 12], [**4**, 2]], [**5**, [0.002, 2, 15], [**PoA1**, 1]]
6	[**6**, [0.0005, 3, 14], [**PoA3**, 1]], [**6**, [0.9002, 4, 11], [**4**, 2]]
7	[**7**, [0.001, 40, 11], [**PoA3**, 1]]
8	[**8**, [0.001, 44, 21], [**7**, 1]], [**8**, [0.0017, 6, 15], [**5**, 2]], [**8**, [0.002, 3, 18], [**5**, 3]], [**8**, [0.0026, 4, 15], [**6**, 1]], [**8**, [0.9004, 5, 12], [**6**, 2]]

The upper bounds of some attributes, such as P_{loss} or jitter, can be directly considered in the MDA. The label-setting procedure can be modified by eliminating labels that do not meet the constraints on the upper bounds. For instance, let's consider the case of an application that requires $P_{loss} \leq 0.002$ and $jitter \leq 40$ ms. The modified MDA gives the output shown in Table 9.3. This revised output contains a reduced number of non-dominated labels due to the inclusion of upper bounds on P_{loss} and jitter. Among this set, for instance, the path with the minimum latency is **UE-PoA2-4-5-8**.

Table 9.3: Output of the MDA algorithm for the example "Graph without MEH State Information" with the constraints $P_{loss} \leq 0.002$ and $jitter \leq 40$ ms.

Node	Non-dominated Labels
UE	[**UE**, [0, 0, 0], ['NULL', 'NULL']]
PoA1	[**UE**, [0.0, 1, 10], [**UE**, 1]]
PoA2	[**PoA2**, [0.0, 3, 12], [**PoA3**, 1]], [**PoA2**, [0.0017, 2, 5], [**UE**, 1]]
PoA3	[**PoA3**, [0.0, 2, 10], [**UE**, 1]]
4	[**4**, [0.0, 4, 15], [**PoA2**, 1]], [**4**, [0.0017, 3, 8], [**PoA2**, 2]]], [**4**, [0.002, 2, 15], [**PoA1**, 1]]
5	[**5**, [0.0, 6, 19], [**4**, 1]], [**5**, [0.0017, 5, 12], [**4**, 2]], [**5**, [0.002, 4, 19], [**4**, 3]]
6	[**6**, [0.0005, 3, 14], [**PoA3**, 1]]
7	[**7**, [0.001, 40, 11], [**PoA3**, 1]]
8	[**8**, [0.0017, 6, 15], [**5**, 2]], [**8**, [0.002, 5, 22], [**5**, 3]]

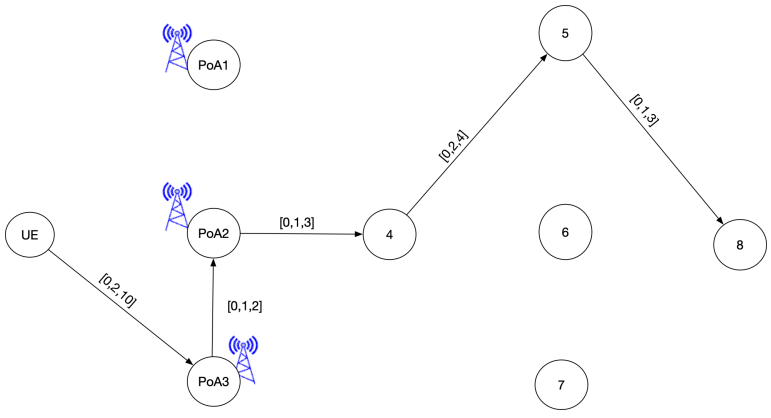


Figure 9.3: Best $(UE, 8)$-path chosen among the set of Table 9.3 using the P_{loss} as high priority attribute.

9.3.3 Graph with MEH State Information

In the second scenario, the MDA is employed on the entire graph depicted in Figure 9.2, which includes the information regarding the performance of MEHs at the application layer. The output of the MDA consists of the collection of all non-dominated paths between the user (represented by node `UE`) and the MEHs that can support the desired application, namely nodes `MEH5A`, `MEH6A`, `MEH7A`, and `MEH8A`.

For each candidate MEH, the cost vector is derived by augmenting the non-dominated set, as presented in Table 9.3, with the additional costs associated with the application layer performance (represented by the red links in Figure 9.2). The outcomes of this process are illustrated in Table 9.4.

Table 9.4: Cost and non-dominated paths for each MEH - Case "Graph with MEH State Information" with the constraints $P_{loss} \leq 0.002$ and $jitter \leq 40$ ms.

MEH	Non-dominated Labels to achieve the network layer of the MEH
MEH5A	[**5**, [0.3, 6, 34], [**4**, 1]], [**5**, [0.3017, 5, 27], [**4**, 2]], [**5**, [0.302, 4, 34], [**4**, 3]]
MEH6A	[**6**, [0.1005, 3, 24], [**PoA3**, 1]]
MEH7A	[**7**, [0.101, 40, 26], [**PoA3**, 1]]
MEH8A	[**8**, [0.3017, 6, 30], [**5**, 2]], [**8**, [0.302, 5, 37], [**5**, 3]]

Depending on the specific application requirements, the selection of the MEH considers various parameters. For instance, in the case of an application prioritizing minimum P_{loss}, the chosen MEH is `MEH6A`. The $(UE, MEH6A)$-path selected from the non-dominated set provided by MDA is depicted in Figure 9.4. In the scenario where the application prioritizes minimum latency while also favoring low P_{loss}, the selected MEH remains `MEH6A`.

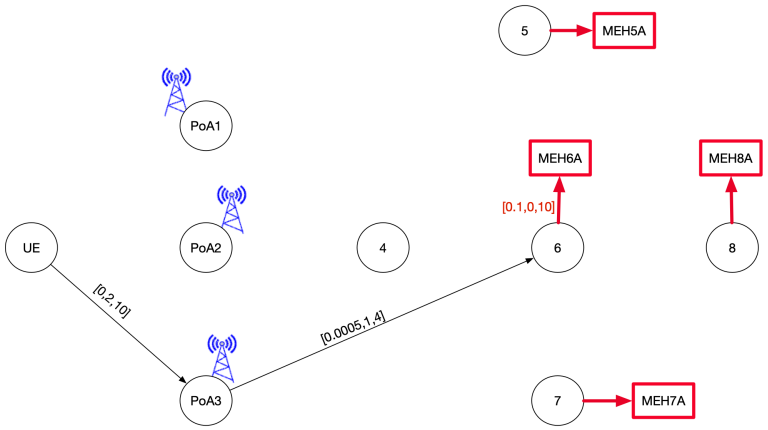


Figure 9.4: The $(UE, MEH6A)$-path derived from the graph shown in Figure 9.2, while adhering to the constraints $P_{loss} \leq 0.002$ and $jitter \leq 40$ ms. The path with lowest P_{loss} is chosen from the set of non-dominated paths shown in Table 9.4.

9.4 Experimental Performance Evaluation

This section thoroughly presents the experimental setup of the testbed. This emphasis is motivated by the need to ensure transparency and reproducibility of our work. Moreover, the experimental setup is instrumental in elucidating the pivotal role of the proposed controller, which stands as one of the key contributions of this paper. To evaluate the developed MDA-based controller, a hybrid (simulative-emulative-experimental) testbed based on AdvantEDGE has been implemented.

AdvantEDGE platform provides an emulated and experimental environment with edge-enabling technologies [236]. The platform runs on Docker and K8s, and provides experimentation with MEC deployment models along with their applications and services. The emulation platform supports several standardized APIs and edge services standardized by the ETSI MEC, including ETSI MEC 013 Location [25], ETSI MEC 012 Radio Network Information [23], ETSI MEC 028 WLAN Information [253], ETSI MEC 011 Edge Platform Application Enablement [27], and ETSI MEC 021 Application Mobility [81]. AdvantEDGE allows the mobility of the UEs within the network by using its own APIs to evaluate the impact on the application performance. The platform allows for configuring the network layer performance in the 5G-MEC architecture through the AdvantEDGE API, which includes settings for latency, jitter, throughput, and packet loss for each link in the emulated network scenario. The platform allows

mobility events, UE movement and mapping of the geo-location of each element. The UE movement can be monitored and visualized by using the Geospatial Subsystem. Furthermore, AdvantEDGE supports the inclusion of MEHs that are not emulated but running on separate devices.

To address performance at the application layer, the scenario incorporates MEHs capable of running and migrating applications as needed. These MEHs form a unified cluster of K8s nodes. Transport-related data is obtained from the AdvantEDGE platform, while application-related information is acquired from the MEC. Interaction with AdvantEDGE allows for the retrieval of transport information, primarily utilizing the location API [25]. During the experimentation phase, the location API is leveraged to track the physical location of UEs within the network, acquiring the necessary graph information as input for MDA. In the presented experiments, the VLC application [239] is utilized, where the VLC client operates on the UE, while the VLC server is hosted on an MEH.

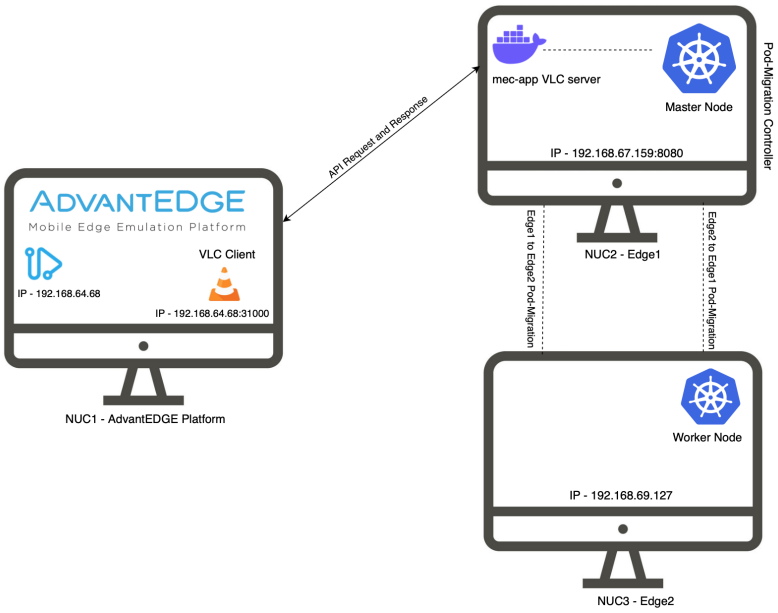


Figure 9.5: The testbed.

Figure 9.5 shows the physical testbed with logical connectivity of the involved elements. The testbed is composed of 3 GIGABYTE(32/512) Intel i7 NUCs. `NUC1` is responsible for running the emulated network scenario implemented with AdvantEDGE, as well as hosting the VLC client of the

UE. The AdvantEDGE platform is installed on a single K8s node running Ubuntu 20.04.4 LTS Operating System (OS). Access to the AdvantEDGE platform GUI is achieved through the IP address 192.168.64.68, enabling the configuration and deployment of the emulated network scenario.

`NUC2` is designated as the MEH responsible for the initial deployment of the MEC App, specifically the VLC server. Conversely, `NUC3` serves as an alternative MEH to which the MEC App can migrate. The choice between `NUC2` and `NUC3` as the migration destination depends on the geographical position of the UE. In the scenario configuration, the external MEC App is associated with the MEH by using the IP address and the port number of the related `NUC2` and `NUC3`. This enables AdvantEDGE to provide support for conducting experiments involving external nodes and applications.

9.4.1 Migrating the MEC App

The K8s cluster, shown in Figure 9.6, comprises a master node and one or more worker nodes. The master node has the control plane functions necessary to coordinate and orchestrate the activities of the worker nodes of the cluster. It plays a crucial role in managing the overall cluster state, scheduling pods, and exposing the API for cluster interactions. The main components of a master node, shown in the figure, are the following:

- The Kube API Server acts as the central control point for interacting with the K8s cluster. It exposes the K8s API, allowing users, administrators, and other components to communicate with the cluster.
- The Controller Manager is responsible for maintaining the desired state of the cluster. It continuously monitors the cluster resources and ensures that the current state matches the desired state defined in the cluster configuration.
- The Scheduler is responsible for assigning pods to nodes in the cluster based on resource requirements, node constraints, and other policies.
- The etcd is a distributed key-value store used by K8s to store the cluster configuration data, including information about nodes, pods, services, and other objects.
- Kube-proxy is a component that runs on each node of the K8s cluster and is responsible for implementing the necessary network routing

for services based on instructions received from the control plane. Specifically, when a pod is migrated, Kube-proxy updates the network configuration of local node to ensure that client traffic destined for the service virtual IP address is correctly routed to the appropriate pods backing the service. A service is a method used in K8s for exposing a network application that runs as one or more pods.

The worker node is responsible for running the actual workloads, including containers and pods. It contributes to the overall execution and management of workloads in the K8s cluster, allowing applications to run and scale efficiently across the distributed environment. To achieve this goal, a worker node implements the following functions shown in the figure:

- Kubelet is an agent that receives instructions from the master node, schedules pods onto the node, and monitors their health and resource usage. It is responsible for managing the state of the node and ensuring that the containers and pods on the node are running as expected.
- The Container Runtime is responsible for running and managing containers. It provides the environment for running application containers and manages their lifecycle, including image management, container creation, starting, stopping, and resource isolation. An example of a container runtime is `containerd`.
- Kube-proxy enables the communication between services and pods within the cluster. It manages network routing, load balancing, and service discovery, allowing applications to communicate with each other and access services seamlessly.
- The Pod is the fundamental unit of deployment in K8s. Pods encapsulate one or more containers and share the same network namespace, storage volumes, and scheduling constraints. The worker node ensures that the containers within the pods are running and handles their lifecycle, resource allocation, and networking.

Referring to this general architecture of a K8s cluster, the `NUC2` implements both control plane functions of a master node and data plane functions of a worker node, while `NUC3` is only a worker node.

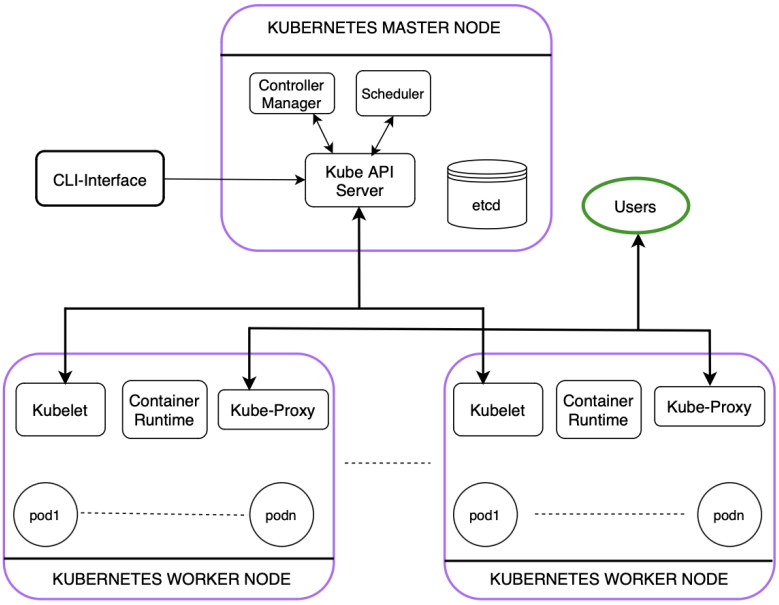


Figure 9.6: K8s cluster and components.

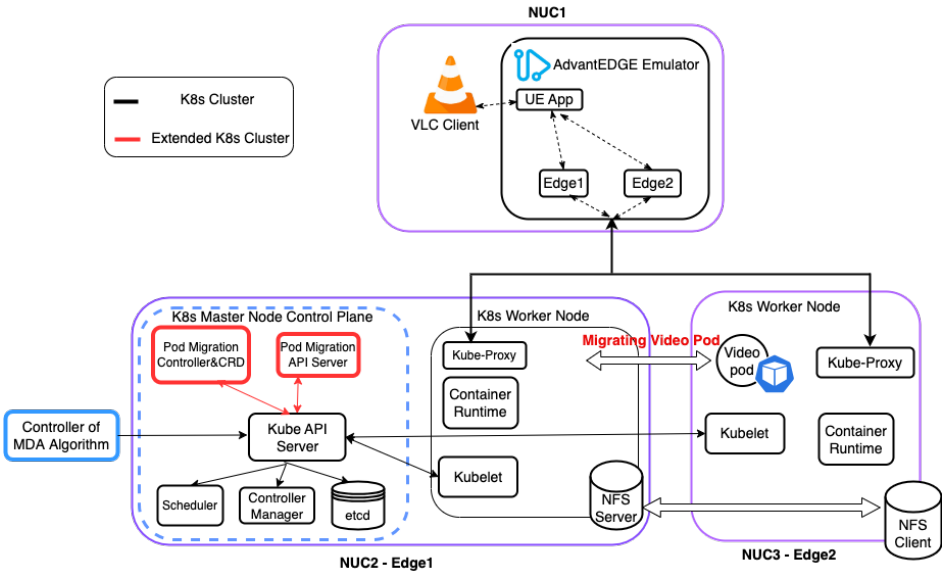


Figure 9.7: Architecture for video pod migration in the experimental testbed.

To enable manual pod migration in K8s, additional elements and configurations need to be incorporated into the cluster. By default, K8s does not support manual pod migration between nodes initiated by the operator. However, it offers built-in mechanisms for pod migration in specific scenarios, such as node draining or when constrained by other factors. In this case, an extended version of K8s is utilized, as referenced in [149], which incorporates the essential features and functionalities needed for smooth and efficient pod migration operations.

In this architecture, the MEC App is implemented in a Docker container and deployed using a pod referred to as `video pod` that supports the VLC server application. The runtime migration is implemented by using the extended K8s version, allowing the runtime pod migration to use the CRIU tool. CRIU enables the checkpointing and live migration of running containers from one node to another within a K8s cluster. When a pod migration is initiated, CRIU captures the state of the running containers, including their memory, file system, and network connections. This captured state is subsequently transferred to the target node, where it is used to restore the containers, ensuring uninterrupted execution. In the testbed, an important aspect is preserving the state of the VLC server by retaining the last transmitted frame of the video stream for each individual user.

Figure 9.7 shows the details of the architecture responsible for the `video pod` migration. `NUC2` and `NUC3` implement the extended K8s version necessary for application migration. This extended version of K8s (in specific the `pod migration API server`) is an element that provides support to the `kubectl-migrate` and `kubectl-checkpoint` commands [149], which have been integrated into `kubectl`, the command-line interface (CLI) tool used to interact with the K8s cluster. It acts as a control plane client and allows users to manage and control various aspects of the cluster. The `pod migration API server` facilitates information exchange between the `pod migration controller`, UE App (i.e., the VLC client), and the K8s nodes. It directs the pod migration from `NUC2` to `NUC3` and vice versa.

During the experiments, the controller developed for testing MDA runs on `NUC1` and interacts with AdvantEDGE. The controller utilizes the APIs of ETSI MEC specifications provided by the platform to acquire UE and network information. This acquired information is utilized by the MDA to determine the need for triggering the migration of the MEC App. When migration becomes necessary, the developed controller sends a request to the Kube API Server, which initiates the migration by sending the request to the `pod migration controller`. This controller generates two essential

commands, namely `kubectl-checkpoint` and `kubectl-migrate`, for the migration process. The `kubectl-checkpoint` command is responsible for creating a checkpoint of the running container and saving its state as files on the respective nodes. Conversely, the `kubectl-migrate` command facilitates the migration of the application within the nodes.

The `pod migration controller` is a component that facilitates the migration of pods from one node to another within a K8s cluster. It manages the process of moving pods while ensuring minimal disruption to the running applications. It is worth noting that this element is not directly connected to the Controller Manager, which primarily manages the deployment and health of nodes and pods. The `pod migration controller` used in this architecture includes Customized Resource Definition (CRD) and a custom controller to monitor the pod migration within the K8s cluster. The `CRD` mechanism supports user-defined data types in K8s and allows the design of the required state, which will be transferred to the target node by the `pod migration controller`. When the migration is triggered, this controller orchestrates the migration process by following a set of steps:

- Designate a specific pod migration controller that will assume the responsibility of overseeing the entire pod migration process.
- Identify the pod to be migrated.
- Prepare the target node by ensuring that it has the necessary resources and dependencies to accommodate the migrated pod. This may involve allocating resources, setting up networking, and preparing the environment.
- Capture the state of the pod on the source node, including its network connections, attached volumes, and other relevant information (in the considered framework, this is done by checkpointing the application). This state information is crucial for preserving the pod functionality during the migration.
- Transfer the captured state from the source node to the target node through the network connecting the two NUCs. This step ensures that the pod state is replicated on the target node.
- Initiate the restoration process on the target node. Once the state transfer is complete. The controller ensures that the pod containers are started, network connections are established, and any necessary

dependencies are met. The pod resumes execution on the target node, seamlessly continuing its tasks.

- Update the Kube API server, reflecting the changes in the pod location and status.

The source node updates its state by removing the migrated pod and reclaiming any previously allocated resources. The `Kube API server` maintains the logs of each step in the application migration process, including the time of the migration request and its completion. It also manages the synchronization between the nodes to ensure proper coordination throughout the migration process.

A shared NFS folder, `/var/lib/kubectl/migrate`, is created and utilized to facilitate the sharing of recorded checkpoint information acquired in the MEH where the MEC App is running. In the testbed, `NUC2` is configured as the NFS server, while the `NUC3` worker node acts as the NFS client.

The successful migration of the MEC App depends on the coordination and interaction of these components, as well as the compatibility and appropriate configuration of the extended K8s version. The use of CRIU and the integration of the extended K8s version enable the checkpointing and migration of running containers, ensuring a smooth transition of the MEC App between nodes within the K8s cluster.

The implementation of MEC and MDA may require a high level of technical expertise and resources. However, in our testbed we do not require any customization of the MEC architecture but the developed controller, which can be seen as part of the MEO, allows to easily integrate MEC and MDA and hides the MDA complexity. The controller collects the system information by exploiting the standard MEC APIs. This information includes data such as UE location, Radio Signal-to-Noise Indicator (RSNI), link performance, and other telemetry data. The information is then delivered to the MDA, which consequently computes a solution. The controller then applies the MDA solution in the system. Additionally, in general ETSI simplifies the implementation of MEC by giving the option of reusing elements of Network Function Virtualization (NFV) to implement the MEC architecture [267].

9.4.2 Network Scenario

The design of the network scenario is motivated by the imperative need to conduct a functional experimental evaluation of the proposed controller proof-of-concept. The chosen scenario is intentionally crafted to encompass a dynamic environment with multiple handovers and mec-app migrations. This deliberate scenario construction aims to stress-test the capabilities and responsiveness of the proposed controller under conditions simulating real-world challenges in contemporary wireless networks.

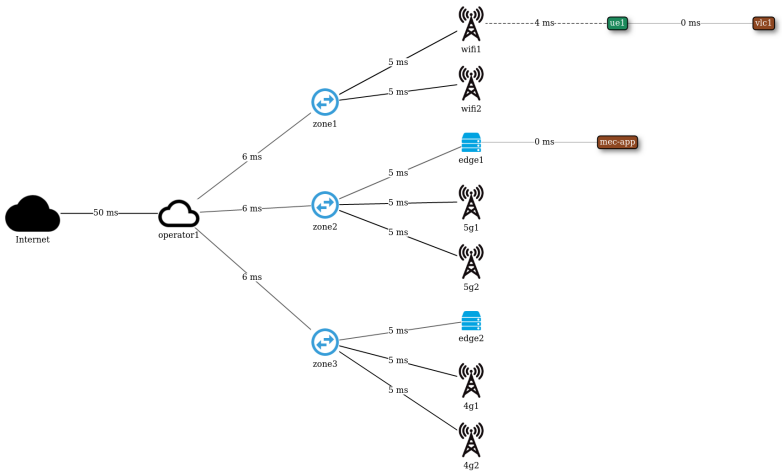


Figure 9.8: Network scenario described by the AdvantEDGE GUI.

Figure 9.8 shows the starting point of the network scenario as depicted in the AdvantEDGE GUI. The scenario consists of a single UE (`ue1`) ocated in `zone1`, while `zone2` and `zone3` contain the emulated MEHs `edge1` and `edge2` respectively. In the testbed, `edge1` is deployed in `NUC2`, whereas `edge2` is deployed in `NUC3`. Each zone is equipped with a different network access technology: `zone1`–WiFi, `zone2`–5G, and `zone3`–4G. Depending on its location, the `ue1` can establish a connection to the MEHs through one of the two PoAs available within each zone. The MEHs are represented by blue boxes in the figure, while the brown boxes indicate the locations of the application elements. In this particular scenario, the application elements consist of the Video Pod (`mec-app`) running in MEH `edge1`, and the VLC

client (`vlc1`) running on `ue1`. The physical UE is depicted as a green box, while the antennas represent the PoAs. In the figure, `Operator1` is the ISP, which provides the IP connectivity through the three access technologies and the IP services supported by the MEC architecture.

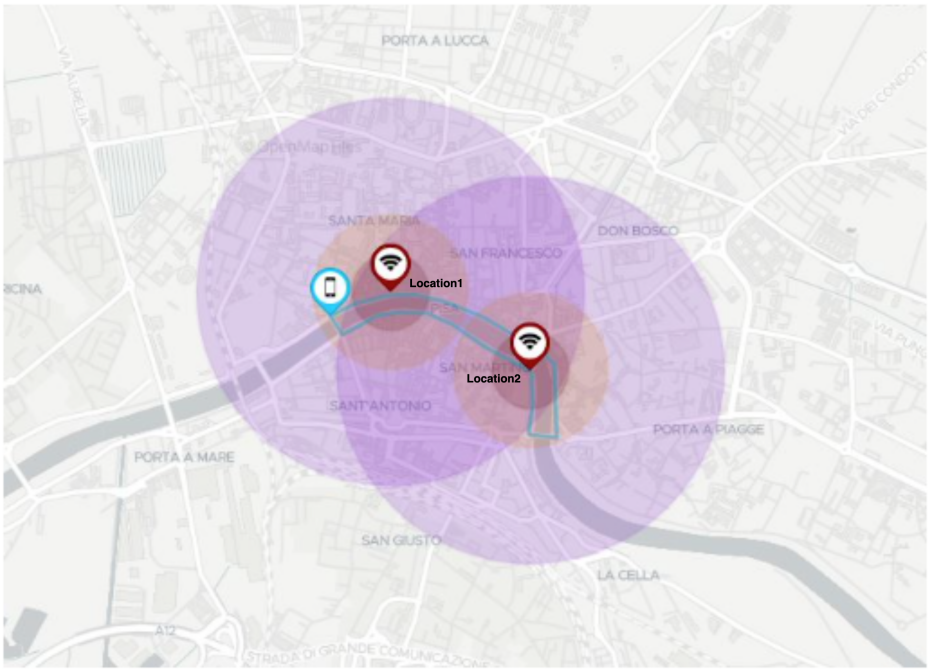


Figure 9.9: Map of the scenario considered in the experimental analysis with AdvantEDGE platform.

The AdvantEDGE platform facilitates the assignment of physical locations for each element depicted in the scenario. The three distinct networking technologies of the PoAs are geographically mapped in different locations, as illustrated in Figure 9.9. The geographical setting represents the area surrounding the Arno River in Pisa. The WiFi PoAs have a coverage radius of 200 meters (indicated in red), while the 5G PoAs span 500 meters (in orange) and the 4G PoAs extend up to 1000 meters (in purple). The blue line in the figure indicates the path followed by `ue1` for the experimental analysis.

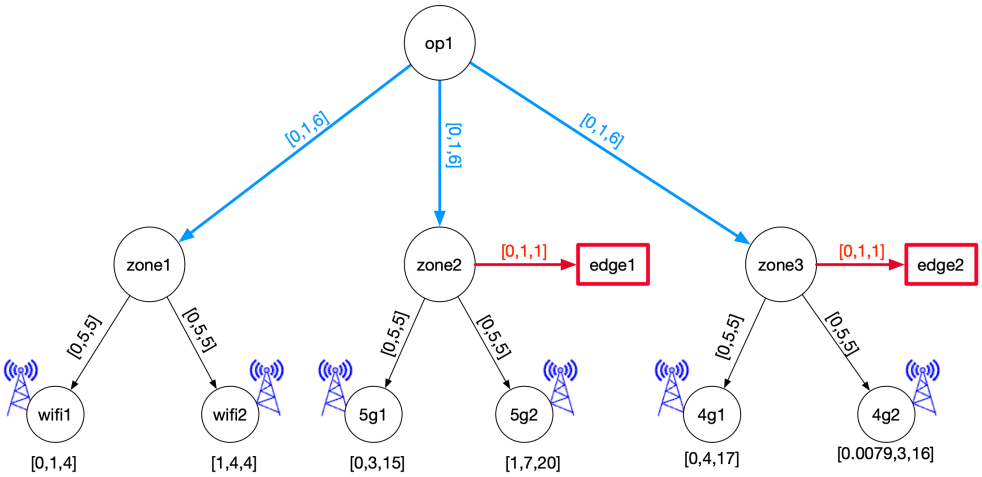


Figure 9.10: Graph depicting the network scenario being analyzed, with each link accompanied by a vector representing the metrics of packet loss probability, jitter, and latency. The blue links operate at a datarate of 2 Mbps, while the black links and wireless connections between the PoAs and `ue1` within the coverage range operate at a datarate of 100 Mbps. The red links connecting the application layer of the MEH and the corresponding network layer have a datarate of 1 Gbps.

9.4.3 Applying MDA

The graph of the considered network scenario is shown in Figure 9.10. In this case as well, the attributes assigned to each link are the packet loss probability, jitter, and latency. The links connecting `op1` with the various zones are assumed to operate at a datarate of 2 Mbps. The black links and wireless connections between the PoAs and `ue1` within the coverage range have a datarate of 100 Mbps. Furthermore, the datarate for the connection between the application layer of the MEH and the corresponding network layer of the supporting node is 1 Gbps, as indicated by the red link in the figure.

During the experimental sessions, the GIS API (`getGeoDataByName`) is utilized to acquire information regarding the physical position of `ue1`, which is necessary for calculating the distance between `ue1` and the various PoAs. This information is employed to determine the set of PoAs capable of providing connectivity to `ue1`. Given the distance between the PoAs and the UE, AdvantEDGE lacks the capability to dynamically compute the datarate of the corresponding wireless links. However, it does provide information regarding the available PoAs, i.e., PoAs that have the UE

within their coverage range. When a PoA is available, the datarate is set to the value manually configured by the user during the setup of the network scenario. The Sandbox API (sendEvent) of AdvantEDGE enables the capability to change the PoA to which `ue1` is connected (i.e. performing the PoA handover), allowing for PoA handover, during runtime. The data required for constructing the graph, including nodes, arcs and their attribute values, can be obtained at runtime by using the APIs of the MEC architecture. To simplify the experimental tests, the attribute values of the arcs are assumed to remain constant throughout the experiments. Therefore, these values have been manually configured in AdvantEDGE.

The test is based on the application of MDA to the dynamic graph generated by using AdvantEDGE information. It is worth noting that the attributes of the link are assumed constant, but the mobility of `ue1` varies the alternative links available to the `ue1` to reach the available MEHs. Depending on `ue1` location, the MDA output gives the selected MEH and the serving PoA of `ue1`, which establish the traffic path in the considered scenario.

Table 9.5 presents the performance parameters of the available paths from the UE to the MEH for all available PoAs and MEHs. The values presented in the table indicate the performance obtained from the MDA output when the UE is located within the coverage range of the respective PoA. The table highlights that the dominant solution is to connect to `5G-1` and utilize MEH `edge1`. Consequently, when the UE is near location 1 and within the coverage range of `5G-1`, `5G-1` and `edge1` are selected. If `5G-1` is not within range, then the predominant solution is `4G-2` with the utilization of MEH `edge2`.

It is worth noting that AdvantEDGE does not consider the control plane procedures for executing handovers between PoAs of the same technology or between different technologies (i.e. multi-Radio Access Technology (multi-RAT) handover). As a result, the delay and certain performance issues (such as packet loss or increased jitter) induced by these procedures are neglected.

9.4.4 Test Execution

The simulation involves the interaction of multiple components: AdvantEDGE, the developed controller with MDA, and K8s. The logic of the developed controller is described by the flow chart shown in Figure 9.11.

Table 9.5: Performance parameters of the path for the experimental tests as a function of the PoA and MEH.

`edge1`-UE	WiFi-1	4G-1	5G-1	WiFI-2	4G-2	5G-2
PLoss (%)	0	0	0	1	0.0079	1
Jitter (ms)	9	12	9	12	11	13
Latency (ms)	22	35	21	22	27	26
Min. Data Rate (Mbps)	2	2	100	2	2	100
`edge2`-UE	WiFi-1	4G-1	5G-1	WiFI-2	4G-2	5G-2
PLoss (%)	0	0	0	1	0.0079	0.1
Jitter (ms)	9	10	11	12	9	15
Latency (ms)	22	23	33	22	22	38
Min. Data Rate (Mbps)	2	100	2	2	100	2

The figure illustrates the initial configuration of the network scenario in AdvantEDGE, including the placement of PoAs, zones, MEH, and the performance parameters of the links. Additionally, the settings for the initial position, speed, and application-related parameters of `ue1` are required. Subsequently, the movement of `ue1` is initiated using the automation feature in the AdvantEDGE GUI, allowing `ue1` to access its application. In this specific case, the VLC client begins displaying the video stream provided by the `mec-app` (i.e., the VLC server) hosted in `edge1`. All this information is loaded by using the AdvantEDGE-specific API known as Sandbox API. The test is prepared for execution once the developed controller subscribes to the ETSI-specific MEC (013) location service API. This subscription is essential for obtaining real-time data necessary to generate the network graph, which serves as input to the MDA algorithm. Specifically, the location of `ue1` is monitored every second, enabling the application of the MDA.

In general, the output of the MDA determines the MEH that will support the UE service `mec-app` and the path to connect `ue1` with the selected MEH. The first aspect may require the migration of the `mec-app`, while the second aspect can modify the path. Due to the simplified network configuration in AdvantEDGE, the traffic path can only be modified through a PoA handover. As previously mentioned, a PoA handover entails the migration of the `mec-app` and vice versa, as indicated by the values in Table 9.5. Therefore, the flow chart in Figure 9.11 solely includes the control for the

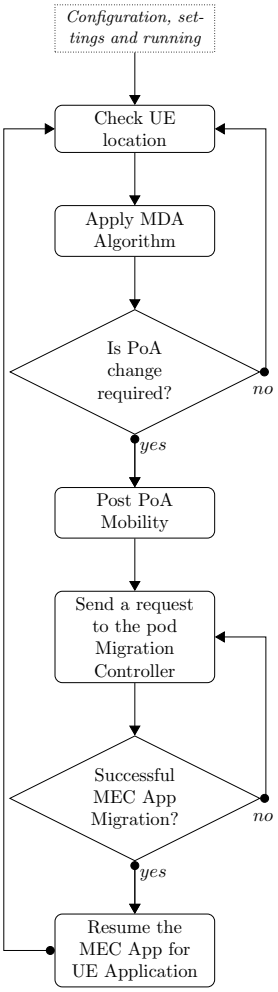


Figure 9.11: Flow chart depicting the logic of the developed controller for the network scenario under consideration. The migration is triggered by the PoA handover, as indicated by the MDA results presented in Table 9.5.

required PoA handover, as this change also involves the migration of the `mec-app`. In general, two separate controls, one for the PoA (or path in the general case) and one for the MEH, are necessary to account for the other cases: i) no PoA handover and no App migration, and ii) PoA handover but no App migration.

In the case where the MDA triggers a PoA handover, two actions are

required. First, the PoA handover is posted through the Sandbox API of AdvantEDGE. This action is necessary to establish the new traffic path between `ue1` and the `mec-app`.

Secondly, if the new path leads to a different MEH, the `mec-app` migration request is sent to the `pod migration controller` via `Kube API server`. Upon receiving this request, the `pod migration controller` creates and assigns a dedicated video controller pod for the migration process. The video controller pod verifies the information of the `mec-app` pod, including the pod name and its running status. Once the `mec-app` pod name is obtained, the `pod migration controller` migrates the pod to the designated MEH using the `kubectl` commands. A new name is assigned to the migrated `mec-app` pod based on the MEH selected by the MDA output. Subsequently, the video controller pod verifies the updated information of the new `mec-app` pod, such as its running status, and relays this information to the `pod migration controller`. Finally, the `pod migration controller` deletes the associated video controller pod and checks the status of the `mec-app` pod in the new MEH using K8s.

The verification process may encounter errors due to various reasons, such as the non-existence or non-running state of the `mec-app` pod in the new MEH, or the absence of the current application state in the shared NFS folder, among others. In such cases, the strategy involves sending a new request to the `pod migration controller` to attempt another migration of the same `mec-app`.

On the contrary, if the control confirms the successful completion of the pod migration, the `mec-app` service is resumed. It is important to note that the information regarding the restart of the `mec-app` service is limited to the VLC server, as K8s cannot verify if `vlc1` maintains the service session. During the experimental tests, it is observed that when the `mec-app` restarts, the `vlc1` application maintains the session, and user experiences minimal service degradation.

9.5 Experimental Results

Experimental tests are conducted on the same testbed, exploring two distinct network scenarios. These scenarios primarily differ in the presence or absence of the MEH `edge2`, while MEH `edge1` is always present. In the first scenario, the aim is to assess the benefits of service migration between the two MEHs when such migration becomes necessary to maintain the

desired service quality. In the second scenario, the focus shifts towards observing the degradation in QoS when the service relies only on MEH `edge1`. In this case, the mobility of `ue1` leads to the PoA handover to maintain the network connection, but the traffic path from the new PoA to MEH `edge1` does not satisfy the QoS requirements.

In both cases, as detailed in subsection 9.4.1, a videostreaming service is considered, implemented by means of a VLC server supported by a pod of the K8s framework. The VLC server streams the video by using the MPEG Transport Stream (MPEG TS) protocol, defined in the ISO/IEC standard 13818-1 [268], over TCP. The video shows "Big Buck Bunny" movie and lasts 597s. The main features of the video are summarized in Table 9.6.

Table 9.6: Features of transmitted video - Audio and video bitrate refer to the average values.

Codec	Video bitrate	Audio bitrate	Width	Height
H.264	3060 kbps	256 kpbs	1920 px	1080 px

The traffic generated by the VLC server is delivered to the `UE-app` (i.e., the VLC client) through the network emulated by AdvantEDGE. The playout buffer of the VLC client is set to 1s. On the data plane, AdvantEDGE adds packet loss, delay jitter, latency and controls each packet transmission time (related to the available link data rate) according to the network characteristics set in the scenario. As a consequence, the quality of the streamed video might be affected to various degrees by the performance of the used network links.

9.5.1 Performance Parameters

During each test, two different classes of performance parameters are collected and analyzed. The first class refers to the parameters obtained from the Grafana dashboard [62] of AdvantEDGE, which are used to acquire data during the emulation. Grafana retrieves metrics such as latency, UL/DL throughput, UL/DL packet loss, and handover events from the InfluxDB database. AdvantEDGE deploys InfluxDB as a pod, creating a dedicated database for the mentioned metrics for each scenario deployment. This information remains available during runtime and until

the scenario is redeployed. The data visualized in Grafana can be exported in `csv` format at the end of the experiment. The following parameters are considered:

- **Latency** is the time a packet takes to be transferred from the ingress to the egress point of AdvantEDGE. Considering a path composed of more than one arc, the overall latency is calculated as the sum of latency values across each arc. AdvantEDGE generates the packet latency values by using a random variable following a Gaussian distribution. Referring to the parameters configured for each link of the AdvantEDGE network scenario, the mean of the distribution represents the latency parameter, while the standard deviation corresponds to the jitter. Each second, Grafana shows the latency obtained by averaging the latency observed by the packets arriving at the egress point in consecutive and non-overlapping time windows of $1ms$.
- **Throughput** is the maximum amount of data that can be transmitted in one second. In AdvantEDGE, the data rate can be configured for each link in the emulated network scenario. The reported value is determined by monitoring the throughput of the designated traffic flow at the egress node of AdvantEDGE. In the considered analysis, the chosen node is `ue1`.

The second class refers to the subjective QoE that is observed by the user during the service. The considered parameter is the **Mean Opinion Score** (MOS), which represents the mean of the absolute score given by the customers according to their satisfaction during the visualization of the video. As recommended by the ITU-T P800 standard [269], an Absolute Category Rating (ACR) is used to score the experience by using a five-point category-judgement, from 1 (Bad) to 5 (Excellent).

9.5.2 Results: Scenario with Two MEHs

In this scenario, the data rate available between the two NUCs implementing the two MEHs is set to $1Gbps$. During the experimental run, the selection of the PoA and MEH used by `ue1` is determined by the MDA algorithm, which receives input from the MEC APIs of AdvantEDGE. The selection process considers the geographical position of `ue1`. Table 9.5 summarizes the performance parameters of the computed path by the MDA when

ue1 is within the coverage of different PoAs. Upon observing the table, it can be noted that ue1 connects to either 5G-1 or 4G-2 based on its geographical position. Furthermore, when ue1 is connected to 5G-1, the MDA recommends utilizing edge2 as the associated MEH. Instead, if the ue1 is connected to 4G-2, the MDA suggests utilizing edge1 as the preferred option. Given these particular MDA results, during the movement of ue1, both the PoA handover and the MEH service migration occur simultaneously. Figure 9.12a displays the end-to-end latency obtained from Grafana, along with the moving average of this time series using a window of 20 samples. As explained in the legend, the vertical lines represent the PoA handovers of ue1 (e.g., 5G-1 or 4G-2), which coincide with the initiation of the mec-app migration. The figure presents a specific time period of the experimental test to illustrate the latency between the MEH and UE, as well as the occurrences of handover events. Table 9.7 displays the maximum, minimum, and average latency values calculated over the entire run.

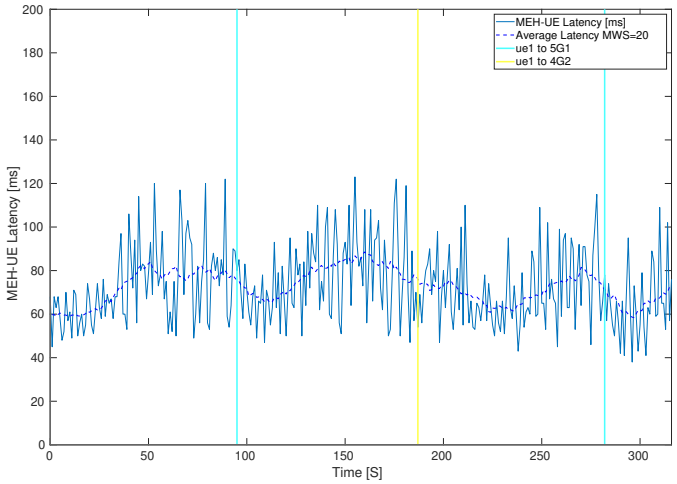
Table 9.7: MEH-UE latency.

Average	Minimum	Maximum
36.35 ms	19.32 ms	61.06 ms

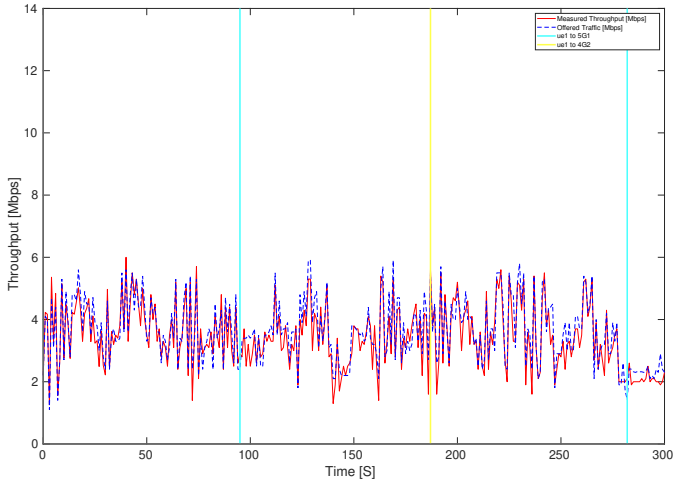
Figure 9.12b shows the measured throughput at ue1, along with the traffic generated by the VLC server and the PoA handover events. The dashed blue lines represent the traffic generated by the VLC server, which was measured using Wireshark [270] under ideal network conditions with no packet loss and high data-rate in each network link.

During the experimental analysis, the ue1 follows the circular track shown in Figure 9.9, with each lap lasting approximately 3 minutes. Within each lap, two migrations occur between MEH edge1 and MEH edge2, and vice versa. The test concludes after observing 100 mec-app migration events. The collected data is then analyzed to evaluate the performance of the K8s platform in terms of mec-app migration time. Table 9.8 presents the statistical parameters for the observed migration times. It can be observed that approximately 5% of the migrations require more than $3.9s$, while the minimum values are below $2s$. By setting the playout buffer of the VLC client to $1s$, the degradation of the video experienced by the end-user is mitigated.

During the migration of the mec-app, often the videostreaming service experiences a temporary freeze on a single image until the migration process



(a) Measured latency, and PoA handover events - Scenario with two MEHs.



(b) Measured throughput, offered traffic, and PoA handover events - Scenario with two MEHs.

Figure 9.12: Performance parameters observed in the scenario with two MEHs.

is completed and the VLC server is restored. In particular, in 11 out of the 100 migration events, the freezing phenomenon of the streamed video

Table 9.8: Statistics on observed `mec-app` migration time in ms.

95% **C.I.**	**Median**	**Min**	**Max**	**95-th percentile**
3015.46 ± 134.533	2866.5	1459	6789	3903.7

was not detected by the user. The video continued to play without any noticeable issues, such as momentary image freezing caused by the need to refill the playout buffer. This result can be primarily attributed to the low migration time and to the size of the playout buffer of the VLC client, which allows for the absorption of video data loss for a period of $1s$. In the remaining migration events, a temporary freezing of the video stream for a few seconds was observed, but the image remained of high quality and free from artifacts. Figure 9.13 presents a sample video image captured during a service migration. The yellow bar and the play button, highlighted by the red ellipses at the bottom of the figure, indicate that the VLC client is buffering the video and the image is temporarily frozen. However, in the worst case, the video restarts after approximately 5 seconds. In summary, the experimental tests result in high-quality video with a MOS score of 4.

During the experiment, the viability of the migration strategy used in the developed testbed is analyzed. The first viability analysis assesses whether the application can initiate and function properly on the target MEH after the migration. The experimental analysis demonstrates that the developed testbed successfully supports application migration while preserving the necessary user state, resulting in a migration process that is nearly transparent to the end user.

The second viability analysis assesses whether the `mec-app` migration can be completed within a time period that allows the application to remain in the target MEH without requiring another migration. To achieve this objective, the interarrival times between consecutive service migration commands are analyzed. The `ue1` follows a circular path with a distance of approximately 0.611 km between two MEHs. Analysis of the data reveals that the migration command is generated with interarrival times ranging from 85 to 87 seconds. These values exceed the maximum observed `mec-app` migration time, which is approximately 6.8 seconds, as presented in Table 9.8. Thus, the migration process is completed before a new migration is triggered.

The size of the data containing the app state information of the client at the VLC server is on the order of kilobytes. These data are crucial for

resuming the streaming at the new MEH after the migration, starting at the exact point (in the considered experiment, the same image and audio) where it was paused in preparation for the migration.

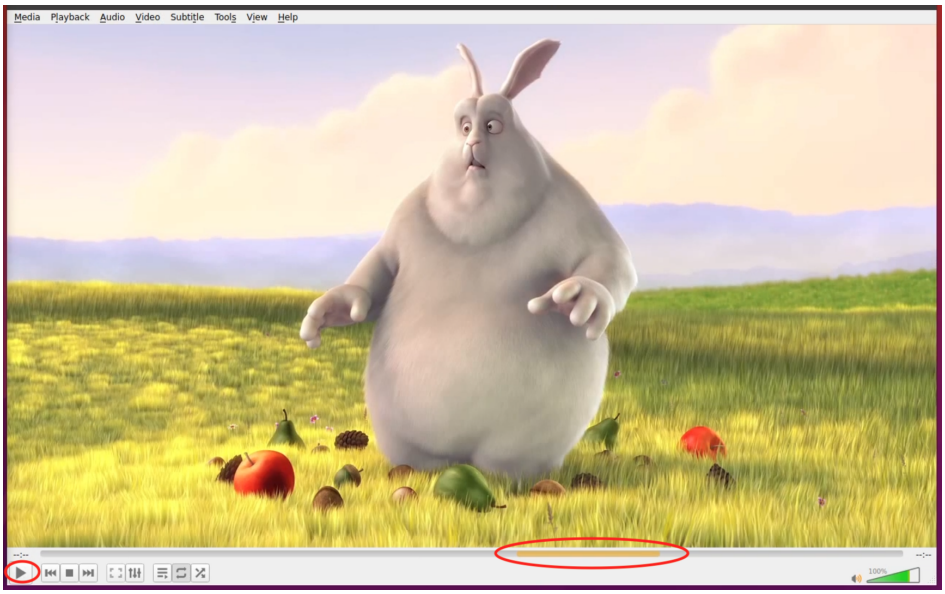
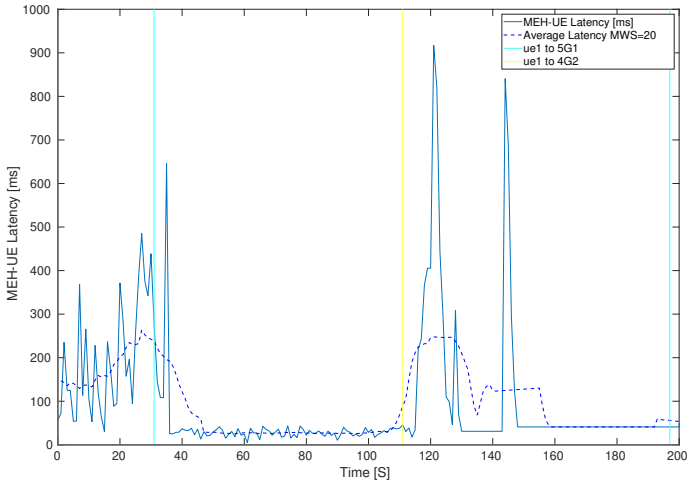


Figure 9.13: Paused image during the `mec-app` migration.

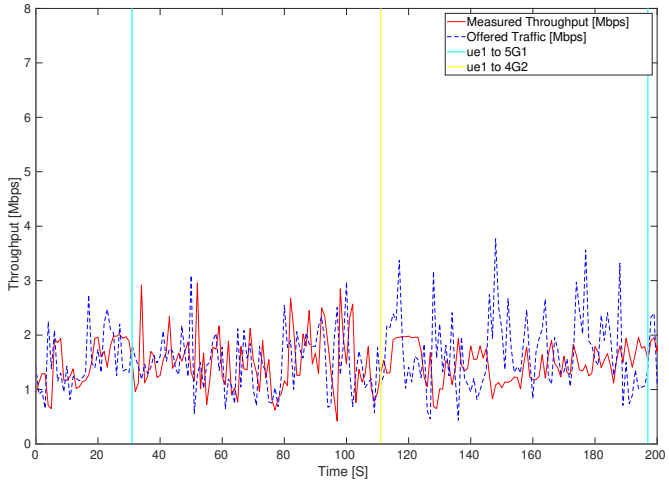
9.5.3 Results: Scenario with One MEH

In this scenario, the VLC service is exclusively provided by the MEH `edge1`. Referring to Table 9.5, the quality of the paths associated with the three available PoA options in location 2 is relatively similar, except for the minimum data rate. Considering this parameter, the strategy that takes into account the constraint on the minimum data rate required to support the application suggests using `5G-2` in this location. MDA can incorporate the constraint on the minimum data rate requirement by excluding all links from the actual graph that have a data rate lower than the minimum requirement. However, for the purpose of comparing the results of this scenario with the previous ones, the selection of PoA `4G-2` is maintained.

Figures 9.14a and 9.14b show the latency and traffic curves, along with the PoA handover events, for a single run of 200s. During this period, three PoA handover events occurred. As observed in the experiment, when the `ue1` is connected to `4G-2`, which is further away from `edge1`, the



(a) Measured latency and PoA handovers - Scenario with one MEH.

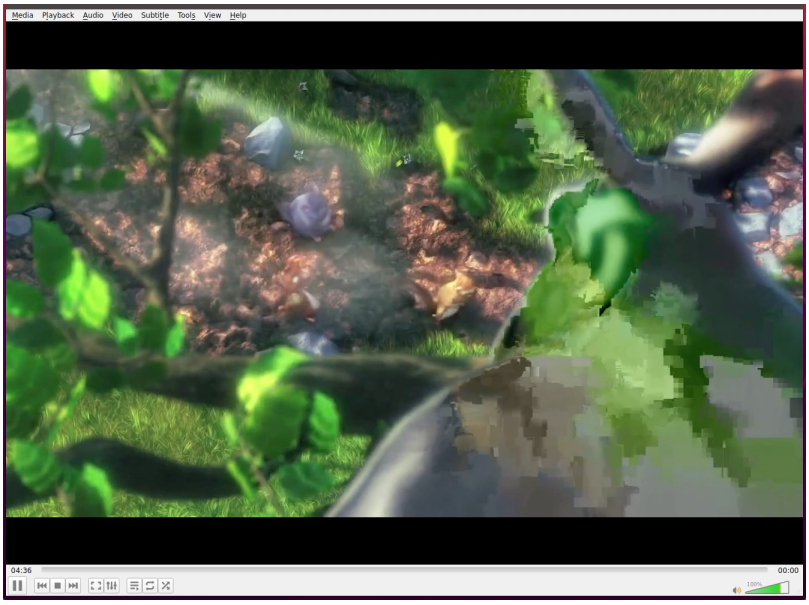


(b) Measured throughput, offered traffic and PoA handovers - Scenario with one MEH.

Figure 9.14: Performance parameters observed in the scenario with one MEH.

latency exhibits significant oscillations with very high values (around 1s). Additionally, the figure reveals that during certain periods, the latency appears to be constant. However, no packets are arriving at the egress point

(a) Scenario with two MEHs.



(b) Scenario with one MEH.

Figure 9.15: Comparison of image quality observed with MEH migration and with one MEH.

of AdvantEDGE during these periods. This phenomenon may be attributed to a link datarate in the network scenario that is lower than the offered

traffic. Consequently, the K8s framework of AdvantEDGE may experience packet loss in the network queuing systems, causing the interruption of data acquisition for latency measurements. Further investigation is required to understand this phenomenon. In these periods, the video quality is severely degraded, with the VLC client displaying a frozen frame of poor quality, as depicted in Figure 9.15b. Instead, when the `ue1` is connected to `5G-1`, the traffic arrives consistently. AdvantEDGE provides the observed latency between the MEH and the `ue1`, which fluctuates based on the link configuration of the scenario. In such cases, the video resumes after a brief period, exhibiting good image quality. Referring to the throughput curves in Figure 9.14b, when `ue1` is connected to PoA `4G-2`, the measured throughput (red curve) is bounded by 2 Mbps. However, there are periods when the offered traffic exceeds this limit, resulting in the very high latency values observed above. When the `ue1` position allows for the use of PoA `5G-1`, the latency returns to the tens of milliseconds range, and the measured throughput at the VLC client aligns with the traffic offered by the VLC server.

To illustrate the impact on the observed quality from the user perspective, Figure 9.15 is presented. This figure enables a comparison of the image quality when the `ue1` is located in location 2. As depicted in Figure 9.15b, the image quality is poor. When comparing this figure with Figure 9.15a, obtained during the test with two MEHs, the enhancement in quality observed by the end-user becomes apparent upon the execution of the `mec-app` migration.

9.6 Related Works and Novelties

Numerous recent works analyze various technical challenges of MEC. Recent surveys, such as [20, 94, 271], summarize the results on three key aspects of the 5G-MEC integrated scenario: security, dependability and performance. Other surveys, such as [73], review the works related to resource allocation in 5G-MEC systems.

The related works can be categorized into two primary classes, although some, like this paper, consider aspects related to both:

- *Protocols and Architecture*: These works focus on presenting architecture solutions, introducing new elements and protocols aimed at reducing service downtime during migration. They also compare

VM-based approaches with container-based ones and define strategies to minimize service downtime. The focus is on the protocol and architecture for service migration.

- *Optimization Algorithms*: These works define optimization problems and algorithms considering specific use-cases, with limited consideration of the protocols and network architecture necessary for deploying the proposed solution.

The methodology employed for performance analysis varies across different works. Some present simulation studies based on ad-hoc models, while others use emulation tools or adopt an experimental approach, integrating a proof-of-concept implementation into a simplified network scenario. This paper utilizes a hybrid (simulative-emulative-experimental) approach. A summary of the novelties of this paper compared to related works, along with a comparison with a selected set of related works, is provided in subsection 9.6.3.

9.6.1 Protocols and Architecture

Focusing on the MEC handover process, Plachy et al. [272] consider that computational resources in the edge are represented by VMs. Thus, the MEC handovers are performed through VM migration (and in general of edge applications migration). This migration implies a rerouting of the traffic to reach the new service location and, in some cases, an exchanging of traffic between the MEHs. Sharghivand et al. [273] propose an Online Service Handoff Mechanism (OSHM) to provide an efficient path dynamically for transferring VM/container from the current serving cloudlet to a nearby cloudlet at the destination of a mobile user. Sarrigiannis et al. [167] explore the implementation of application and VNF migration within an MEC-enabled 5G framework to improve resource optimization and accommodate application-specific demands. Application migration is triggered when MEC resources are depleted, involving the relocation of VMs to better utilize computing resources. VNF migration complements this by reconfiguring network components to address heightened 5G application requirements, especially during periods of increased network traffic.

The volume of traffic exchanged between the source and target MEH depends on the type of migrations (stateless or stateful [274]) and proactive strategies aimed at minimizing the time required to complete the MEC

handover. For instance, Machen et al. [274] introduce a layered framework for migrating active service applications encapsulated in either VMs or containers. This layered approach significantly reduces service downtime. However, automated orchestration is crucial to implement mechanisms that deploy applications at optimal locations and, when necessary, relocate them to meet QoS requirements. Fondo et al. [146] describe an architecture implemented in an experiment demonstrating how Open Source MANO (OSM) can automate the relocation of a video processing application aiding drivers in recalling the latest traffic sign viewed. They propose to add two new components: the first one maintains the state of applications when deployed at a new location, and the second one enables OSM to manage the Open Network Edge Services Software (OpenNESS) edge platform. Wadatkar et al. [275] present a performance evaluation study of a migration technique based on Docker and K8s.

The migration of containerized MEC applications is a key research challenge for supporting low-latency services and for efficient network resource utilization. The migration is necessary to maintain service proximity, reducing the distance between the end user and the application. Different works propose migration strategies [276][277] and experimental comparisons of them [259]. However, for migrating MEC applications, pod migration might be preferred over container migration due to the orchestration and management functions given by K8s. Indeed, K8s provide application portability within the cluster that neglects the complexity of the underlying infrastructure and network condition configurations by offering services such as load balancing, service discovery and fault tolerance. K8s has a default behavior of rescheduling or evicting pods to healthy nodes in the event of a node failure, ensuring the continuity of applications. However, K8s does not inherently handle the check-pointing or preservation of the application state during this process. Consequently, when a pod is replaced by a new instance, the new pod starts somewhere else and does not retain any information or state from the previous pod. To cope with this issue, Schrettenbrunner [278] introduces a migration controller and provides a prototype implementation that demonstrates the feasibility of the proposed approach. Junior et al. [150] present a pod migration mechanism in K8s that enables seamless migration of pods between nodes within a geo-distributed environment. The migration process involves stopping and check-pointing the pod memory state and system-level resources, transferring the checkpoint data to the destination node, and restarting a new pod from the checkpoint. The mechanism implies that the application

needs to modify how it handles and persists in-memory state to ensure its preservation during pod migration. Tran et al. [279] proposes a stateful service migration mechanism that extends the capabilities of K8s by leveraging the Checkpoint/Restore In Userspace (CRIU) project [280] and the Container Runtime Interface (CRI) extension [281]. Their work focuses on enhancing fault tolerance and ensuring the high availability of containerized services by considering both the storage state and in-memory state during migration.

Shah et al. [164] propose an architecture integrating SDN and MEC, capitalizing on SDN for end-to-end mobility and QoS. The architecture was validated through V2X simulations using Mininet-WiFi and Docker emulators. The work addresses service migration issues between MEC and introduces DRS for relocating MEC applications with minimal downtime. Concerning application state migration, Docker volumes are employed, primarily focusing on local container data storage, potentially lacking orchestration for multi-node sharing capabilities. The lack of orchestration could lead to extended periods of downtime. The evolution of this work is presented in [35], where the authors suggest a centralized network and MEC server resource coordinator, utilizing SDN orchestration to manage limited resources in highly mobile environments like V2X. Fondo et al. [165] explore the implementation of an SDN solution for dynamically and transparently relocating communication endpoints using containers within cellular networks. This approach ensures session continuity and reduced latency, especially in congested network conditions.

Kubernetes Role in ETSI-MEC Framework

K8s stands out as the primary container orchestration platform in the contemporary networking landscape. Numerous initiatives are currently engaged in deploying and evolving K8s to meet the functionalities outlined by the ETSI-MEC framework. CAICT et al. [282] extensively discuss the capabilities of the EdgeInfra solution, which manages applications through K8s to enhance Container Network Interfaces (CNIs) and ensures service isolation in Telecom. They underscore the importance of clearly defining the requisites for various industry applications, ideally as K8s templates, to guarantee deterministic 5G capabilities and resource allocation. Other solutions, like EdgeGallery [66] and OpenSigma, also leverage K8s as their edge infrastructure. Martínez-Casanueva, et al. [283] propose an edge computing design based on K8s and Helm, offering function blocks and

APIs as defined by ETSI. The prototype demonstrates the feasibility of a lightweight MEC platform. ETSI proposes two white papers. The first [70] emphasizes the importance of application packaging and runtime environments, such as VMs, Docker containers, or K8s templates within the MEC system. The other [284] discusses MEC system support for edge-native designs, with the Edge Multi Cluster Orchestrator (EMCO) of Linux Foundation proposing the use of K8s clusters for scaling geo-distributed applications and network functions for telco solutions. Escaleira et al. [285] demonstrate the efficiency and viability of integrating K8s within the MEC system, following the ETSI standardized framework. The need for the rapid development of a fully operational MEC infrastructure by seamlessly scaling K8s cluster nodes showcases the potential for K8s to play a pivotal role in MEC architecture. [286] proposes different existing solutions for function mapping within the ETSI MEC framework, where K8s may assume the responsibilities of VIM. Slamnik-Krijestorac et al. [137] conducted a study based on container-based service deployment and established a benchmark for MANO solutions in the MEC context. Barrachina et al. [138] propose the MARSAL MEC framework, which leverages a subset of ETSI MEC/NFV where K8s assumes the role of VIM. Bolettieri et al. [139] demonstrate a novel slicing architecture where orchestration and slicing for MEC applications benefit from using K8s and Helm technologies.

9.6.2 Optimization Algorithms

As concerning the description of algorithms for service placement and routing, Poularakis et al. [287] focus on joint service placement and request routing problem in a MEC multi-cell scenario with multiple constraints, aiming to minimize the load of the centralized cloud. A robustness-aware VNF placement and request scheduling scheme is presented in [288], while a model considering the resource allocation for services, the traffic management of requests, and the path arrangement for data delivery is presented in [289]. The goal of the model is to investigate and quantify the relationship between the performance and cost of the edge-based service provisioning system. Considering the multi-service-provider MEC system, the joint service placement and request routing problem has been analyzed in [290]. In this case, the authors consider that a service provider prefers to use edge servers deployed by itself instead of others, which not only improves service quality but also reduces processing costs. The service placement and request scheduling strategies directly affect the revenue of service providers. A

recent evolution of this work proposes to federate geographically proximate edge servers to form a logically centralized resource pool [291]. However, the optimization of such systems is challenging. Gohar et al. [292] and Sarah et al. [293] consider an intermediate entity, slice broker, that buys virtual resources from infrastructure providers and sells network slices to slice tenants. In a MEC-cloud scenario, the broker selects the MEH and datacenters (and the related resource configuration) and allocates the virtual network functions composing the network slice. Mason et al. [294, 295] consider that the selection of MEH and datacenters has been already made and they dynamically select the data and network resources to allocate to each slice in order to maximize the user experience.

The need of adjusting the service placement and request scheduling is discussed by Farhadi et al. [265], referring to data-intensive applications, such as video analytics, machine learning (ML) tasks. The authors show that, due to time-varying demands, the code and data placement need to be adjusted over time, which raises concerns about system stability and operation cost. They address these issues by proposing a two-time-scale framework that jointly optimizes service (code and data) placement and request scheduling while considering storage, communication, computation, and budget constraints. Anwar et al. [161] propose distributed traffic steering by distinguishing between two distinct types of network elements, namely MEHs and routers. They establish the equivalence between solving the Shortest Path problem, which minimizes the cost derived from the sum of latency and the inverse of available bandwidth, and the Pareto optimal path obtained through multi-objective minimization of latency and the inverse of available bandwidth. The proposed approach yields improved results, incorporating Pareto optimality considerations for minimizing various criteria.

Rodrigues et al. [163] propose a deployment policy for edge servers based on k-means clustering and particle swarm optimization to reduce operational costs and service delays. On the other hand, Cao et al. [162] investigate the edge server placement problem, considering the heterogeneity of servers and the fairness of response time. They propose an approach with both offline and online stages. The work by Doan et al. [166] focuses on optimizing MEC state transfer, considering factors such as optimality, latency, and communication awareness. The goal is to minimize migration costs under various constraints. They introduce the FAST (Flexible And Low-Latency State Transfer) framework, applicable to large-scale networks, and efficiently implement a Tabu search algorithm. This algorithm iteratively explores

the search space, avoids revisiting solutions using a Tabu list, and updates solutions while considering constraints and an aspiration criterion. Ayimba et al. [168] focus on developing a robust controller for Cooperative Adaptive Cruise Control (CACC) of connected cars through MEC in cellular networks. This addresses low-latency connection maintenance and platoon switching. The study introduces a Q-Learning algorithm for network adaptation and evaluates the performance of the migration scheme in comparison to a state-of-the-art scheme. Liu et al. [296] tackle the task migration problem in MEC, introducing a distributed task migration algorithm using Counterfactual Multi-Agent (COMA) reinforcement learning. The primary objective of this algorithm is to minimize the average task completion time while adhering to a migration energy budget.

9.6.3 Summary of the Novelties

Regarding the architecture aspect, the novelty of this paper lies in the utilization of the pod migration framework [149] to conduct an experimental performance analysis of the proposed MDA. To accomplish this, a controller based on the MDA approach is developed, enabling seamless interaction with the components of the pod migration framework. This integration facilitates the implementation of the MDA decision for application migration within the testbed, thereby enabling the experimental evaluation of performance at both the network layer and the application layer.

Regarding the optimization algorithm, the novelty of the study proposed in this paper is that the proposed multi-objective technique can take into account more than two metrics simultaneously. The result is more general because packet loss, bandwidth, latency, and other performance metrics (e.g. power consumption, security level etc.) can be considered simultaneously. Furthermore, the proposed scheme for generating the graph containing information on the performance at the application layer and at the network layer allows to jointly find the solutions to both the traffic path computation problem and the MEH selection problem. Moreover, this paper explores the performance at the network layer as well as the enhancement of user experience through the utilization of the proposed scheme in a hybrid testbed with an ad-hoc controller.

Table 9.9 and 9.10 offers a comparison analysis of a selected set of works in the field related to our contribution. The table outlines the methodology of the study, the contribution, and the pros and cons with respect to our

work. Except for the works in the first row, the others do not adhere to ETSI-MEC architecture.

9.7 Conclusions

The paper addressed the joint selection of the MEH and the path between the UE and the MEH. The paper found the solution to the addressed problem by using MDA through a proposed procedure to create a graph that is able to consider both network-layer and application-layer metrics. The performance of MDA has been evaluated by implementing a hybrid testbed, which is able to migrate a VideoLan application between two MEHs. The testbed is based on AdvantEDGE, which is able to simulate the UE mobility and the radio link, to emulate the network and the MEC APIs, and to experiment the VLC client. Moreover, the testbed includes K8s, which is used to support the migration of the VLC server pod between two actual MEHs. Finally, the paper added in the testbed a controller in order to integrate MDA with the 5G-MEC system. Two evaluations have been performed in the testbed: one with two MEHs; another one with only one MEH. In the evaluations, two network performance parameters (latency and throughput) and the user experience (MOS) have been considered. The results of the scenario with two MEHs demonstrate that MDA is able to perform the migration with a limited impact on the network performance and user experience. Instead, the results of the scenario with only one MEH emphasize that the absence of migration would result in a significant decline in the user experience.

Table 9.9: Comparison with Selected Existing Works (Part 1)

Ref	Methodology	Contribution	Comparative analysis with our work
[35, 164]	Algorithm and Architecture. SUMO Simulation with Mininet-WiFi and Docker	SDN-MEC integration and four modules address mobility, including performance monitoring, service classification, and SDN-assisted traffic steering.	+ Address diverse service requirements and resource management effectively - Lack support for multi-access technology - Overlook multiple objectives (prioritizing latency and bandwidth) - Limited applicability of migration and orchestration to customized MEC environments - Potential portability issues with Docker volume usage for transferring application state
[163]	Algorithm based on ML	Deployment policy based on k-means clustering and particle swarm optimization	+ Utilizes multiple agents for cloudlet deployment and selection - No consideration for network selection and application migration - No implementation and experimental analysis
[162]	Algorithm. Offline and Online stage	Edge server placement considering server heterogeneity and response time fairness	+ Considers edge server placement - No relation to the ETSI-MEC framework - Addresses server selection during user mobility but overlooks application migration - Only simulation analysis, no implementation in a testbed
[166]	Algorithm. Mininet-WiFi and Floodlight SDN controller	MEC state transfer optimization by using FAST framework with Tabu search algorithm	- No consideration of ETSI MEC APIs - No consideration for multi RAT, network selection, and handover - No defined scheme for the execution of application state transfer

Table 9.10: Comparison with Selected Existing Works (Part 2)

Ref	Methodology	Contribution	Comparative analysis with our work
[165]	Experiment with ONOS SDN controller and Docker	SDN solution for dynamic relocation of communication endpoints	- Migration of containerized applications from edge to core without edge host selection - Not aligned with the ETSI MEC framework
[167]	Experiment using Openstack, Linux container, OSM and OAI	Network and MEC resource allocation	- No consideration for network handover, multi-objective selection, MEC path selection, and ETSI-MEC support - Possible hypervisor dependencies with VM migration
[161]	Algorithm. Simulation	Resolve scalability for large MEC network and Multi-objective Optimization	- Selection of the shortest dynamic path for the edge server and router only accounting delay and bandwidth - The selection process overlooks migration based on simulated analysis and is not in line with ETSI MEC APIs
[168]	Algorithm. SUMO Simulator and Reinforcement learning	Development of a robust controller for CACC through MEC	+ Addresses placement issue - No consideration for multi RAT - Q-learning algorithm overlooks multi-objective path selection - Simulated application migration lacks practical validation and decision impact demonstration
[296]	Algorithm with COMA reinforcement learning	Distribute task migration based on multi-agent policy	+ Considers a multi-users scenario - Exclusively designed for task migration, neglects optimal path selection for MEC, lacks network handover - No practical validation for task migration

Paper 5: MigraMEC: Hybrid Testbed for MEC App Migration

MigraMEC: Hybrid Testbed for MEC App Migration

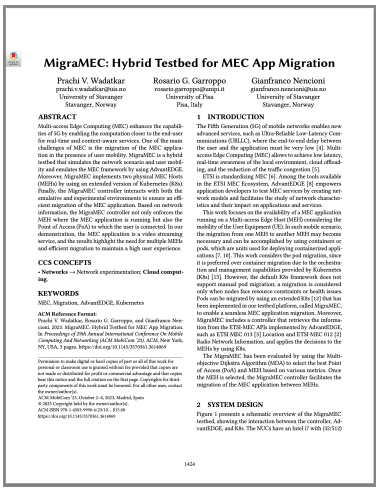
Prachi V. Wadatkar[1,2], Rosario G. Garroppo[2], Gianfranco Nencioni[1]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger

[2] Department of Information Engineering, University of Pisa

Published in the Proceedings of the 29th Annual International Conference on Mobile Computing and Networking (MobiCom), 2023

https://dl.acm.org/doi/10.1145/3570361.3614069



MigraMEC: Hybrid Testbed for MEC App Migration

Prachi V. Wadatkar, prachi.v.wadatkar@uis.no, University of Stavanger, Stavanger, Norway
Rosario G. Garroppo, rosario.garroppo@unipi.it, University of Pisa, Pisa, Italy
Gianfranco Nencioni, gianfranco.nencioni@uis.no, University of Stavanger, Stavanger, Norway

ABSTRACT

Multi-access Edge Computing (MEC) enhances the capabilities of 5G by enabling the computation closer to the end-user for real-time and context-aware services. One of the main challenges of MEC is the migration of the MEC application in the presence of user mobility. MigraMEC is a hybrid testbed that simulates the network scenario and user mobility and emulates the MEC framework by using AdvantEDGE. Moreover, MigraMEC implements two physical MEC Hosts (MEHs) by using an extended version of Kubernetes (K8s). Finally, the MigraMEC controller interacts with both the emulative and experimental environments to ensure an efficient migration of the MEC application. Based on network information, the MigraMEC controller not only enforces the MEH where the MEC application is running but also the Point of Access (PoA) to which the user is connected. In our demonstration, the MEC application is a video streaming service, and the results highlight the need for multiple MEHs and efficient migration to maintain a high user experience.

CCS CONCEPTS

• Networks → Network experimentation; Cloud computing.

KEYWORDS

MEC, Migration, AdvantEDGE, Kubernetes

ACM Reference Format:

Prachi V. Wadatkar, Rosario G. Garroppo, and Gianfranco Nencioni. 2023. MigraMEC: Hybrid Testbed for MEC App Migration. In *Proceedings of 29th Annual International Conference On Mobile Computing And Networking (ACM MobiCom '23), ACM, New York, NY, USA, 3 pages.* https://doi.org/10.1145/3570361.3614069

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
ACM MobiCom '23, October 2–6, 2023, Madrid, Spain
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9990-6/23/0...$15.00
https://doi.org/10.1145/3570361.3614069

1 INTRODUCTION

The Fifth Generation (5G) of mobile networks enables new advanced services, such as Ultra-Reliable Low-Latency Communications (URLLC), where the end-to-end delay between the user and the application must be very low [4]. Multi-access Edge Computing (MEC) allows to achieve low latency, real-time awareness of the local environment, cloud offloading, and the reduction of the traffic congestion [5].

ETSI is standardizing MEC [6]. Among the tools available in the ETSI MEC Ecosystem, AdvantEDGE [8] empowers application developers to test MEC services by creating network models and facilitates the study of network characteristics and their impact on applications and services.

This work focuses on the availability of a MEC application running on a Multi-access Edge Host (MEH) considering the mobility of the User Equipment (UE). In such mobile scenario, the migration from one MEH to another MEH may become necessary and can be accomplished by using containers or pods, which are units used for deploying containerized applications [7, 10]. This work considers the pod migration, since it is preferred over container migration due to the orchestration and management capabilities provided by Kubernetes (K8s) [15]. However, the default K8s framework does not support manual pod migration; a migration is considered only when nodes face resource constraints or health issues. Pods can be migrated by using an extended K8s [12] that has been implemented in our testbed platform, called MigraMEC, to enable a seamless MEC application migration. Moreover, MigraMEC includes a controller that retrieves the information from the ETSI-MEC APIs implemented by AdvantEDGE, such as ETSI MEC 013 [3] Location and ETSI MEC 012 [2] Radio Network Information, and applies the decisions to the MEHs by using K8s.

The MigraMEC has been evaluated by using the Multi-objective Dijkstra Algorithm (MDA) to select the best Point of Access (PoA) and MEH based on various metrics. Once the MEH is selected, the MigraMEC controller facilitates the migration of the MEC application between MEHs.

2 SYSTEM DESIGN

Figure 1 presents a schematic overview of the MigraMEC testbed, showing the interaction between the controller, AdvantEDGE, and K8s. The NUCs have an Intel i7 with (32/512)

1424

Abstract:
Multi-access Edge Computing (MEC) enhances the capabilities of 5G by enabling the computation closer to the end-user for real-time and context-aware services. One of the main challenges of MEC is the migration of the MEC application in the presence of user mobility. MigraMEC is a hybrid testbed that simulates the network scenario and user mobility and emulates the MEC framework by using AdvantEDGE. Moreover, MigraMEC implements two physical MEC Hosts (MEHs) by using an extended version of Kubernetes (K8s). Finally, the MigraMEC controller interacts with both the emulative and experimental environments to ensure an efficient migration of the MEC application. Based on network information, the MigraMEC controller not only enforces the MEH where the MEC application is running but also the Point of Access (PoA) to which the user is connected. In our demonstration, the MEC application is a video streaming service, and the results highlight the need for multiple MEHs and efficient migration to maintain a high user experience.

10.1 Introduction

The Fifth Generation (5G) of mobile networks enables new advanced services, such as Ultra-Reliable Low-Latency Communications (URLLC), where the end-to-end delay between the user and the application must be very low [3]. Multi-access Edge Computing (MEC) allows to achieve low latency, real-time awareness of the local environment, cloud offloading, and the reduction of the traffic congestion [297].

ETSI is standardizing MEC [17]. Among the tools available in the ETSI MEC Ecosystem, AdvantEDGE [236] empowers application developers to test MEC services by creating network models and facilitates the study of network characteristics and their impact on applications and services.

This work focuses on the availability of a MEC application running on a Multi-access Edge Host (MEH) considering the mobility of the User Equipment (UE). In such mobile scenario, the migration from one MEH to another MEH may become necessary and can be accomplished by using containers or pods, which are units used for deploying containerized applications [259, 278]. This work considers the pod migration, since it is preferred over container migration due to the orchestration and management capabilities provided by Kubernetes (K8s) [53]. However, the default K8s framework does not support manual pod migration; a migration is considered only when nodes face resource constraints or health issues. Pods can be migrated by using an extended K8s [149] that has been implemented in our testbed platform, called MigraMEC, to enable a seamless MEC application migration. Moreover, MigraMEC includes a controller that retrieves the information from the ETSI-MEC APIs implemented by AdvantEDGE, such as ETSI MEC 013 [25] Location and ETSI MEC 012 [23] Radio Network Information, and applies the decisions to the MEHs by using K8s.

The MigraMEC has been evaluated by using the Multi-objective Dijkstra Algorithm (MDA) to select the best Point of Access (PoA) and MEH based on various metrics. Once the MEH is selected, the MigraMEC controller facilitates the migration of the MEC application between MEHs.

10.2 System Design

Figure 10.1 presents a schematic overview of the MigraMEC testbed, showing the interaction between the controller, AdvantEDGE, and K8s. The

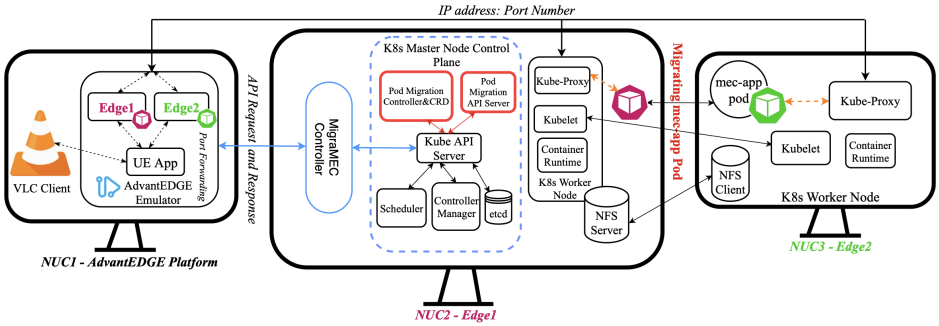


Figure 10.1: Schematic Representation of the MigraMEC Testbed

NUCs have an Intel i7 with (32/512) GIGABYTE. The AdvantEDGE platform is deployed on NUC1, which also serves as the host for a user VideoLAN Client (VLC) [239]. Initially, the MEC application, named `mec-app` and which is a VLC server, is deployed on NUC2 (MEH named `Edge1`) and later migrated to NUC3 (MEH named `Edge2`) based on the location of the UE. NUC2 also serves as the master node within the K8s cluster. Instead, NUC3 is a worker node within the same cluster.

In addition to the default K8s cluster framework, NUC2 and NUC3 incorporates extended version of K8s and Container Runtime Interface (CRI) [281], which is supported by CRIU project [280]. Extended K8s and CRI version is necessary to facilitate the seamless pod migration. The `mec-app` is a VLC server application deployed as a pod that runs within a container. By utilizing K8s service and Kube-proxy mapping, the `mec-app` is made accessible to end-users (in this case, mapped to the AdvantEDGE emulator) through an IP address and port number. `Pod-migration controller` and API server are part of the extended K8s framework that controls and tracks the pod migration activities, respectively. Kube API server interacts with K8s elements and operator (in this case, MigraMEC controller). The NFS server and client are used to share the pod checkpoint information during the migration. The controller manager, scheduler, etcd, are default elements of the K8s cluster responsible for node control, pod scheduling activities, and functioning as a database for cluster information, including authentication keys.

10.3 Demonstration

The considered network scenario consists of two sets of `WiFi`, `5G`, and `4G` PoAs with coverage radius of 200, 500, 1000 meters, respectively. Network characteristics, such as latency, jitter and packet loss, have been set. Furthermore, PoAs and UE have their geographical locations and an actual map representation can be created by AdvantEDGE.

During the demonstration, we will enable the UE mobility by using AdvantEDGE automation. We will show how MDA will select the best PoA and MEH based on the UE location.

Our demonstration focuses on evaluating the performance of the system through two experimental tests. First test corresponds to having only single MEH (i.e., `Edge1`). The absence of an MEH (i.e., `Edge2`) noticeably degrades the video quality, resulting in visible blurriness, while VLC statistical values indicate the loss of frames. Moreover, the absence of an MEH significantly affects the latency and throughput between the MEH and UE, especially when compared to the VLC server throughput. This impact is primarily due to the longer path between the UE and MEH, resulting in increased latency and decreased throughput for MEH-UE communication.

Having two MEHs that support the `mec-app` migration showcased superior performance compared to a single MEH. During UE movement, the `mec-app` was seamlessly relocated to the nearest MEH depending on UE location, resulting in an enhanced Quality of Experience (QoE) for the user. MEH-UE latency and throughput were optimized in comparison to the single MEH test. While migrating the `mec-app` between MEHs, there was a brief buffering period for video frames as the `mec-app` transitioned and restoration took place. However, the average buffering time was less than 3 seconds, considering that the VLC client is typically set with a default buffering value of 1 second.

In the future, the MigraMEC controller will be further developed to not only retrieve network information from AdvantEDGE, but also computing information from K8s.

Bibliography

[1] ETSI - 5G technologies. https://www.etsi.org/technologies/5g. Accessed: April 12, 2024.

[2] International Telecommunication Union (ITU). Forging paths to 5g. https://www.itu.int/en/itunews/Documents/2017/2017-02/2017_ITUNews02-en.pdf, 2017. Online; accessed 12-04-2024.

[3] 3GPP. 3gpp ts 23.501 v16.11.0, "3rd generation partnership project; technical specification group services and system aspects; system architecture for the 5g system; stage 2 (release 16). *3GPP Technical Specifications*, 2021 December.

[4] M Series. Minimum requirements related to technical performance for imt-2020 radio interface (s). *Report*, pages 2410–0, 2017.

[5] International Telecommunication Union (ITU). Recommendation itu-r m.2083-0 (09/2015) imt vision – framework and overall objectives of the future development of imt for 2020 and beyond. https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf, 2015. Online; accessed 12-04-2024.

[6] Jorge Navarro-Ortiz, Pablo Romero-Diaz, Sandra Sendra, Pablo Ameigeiras, Juan J Ramos-Munoz, and Juan M Lopez-Soler. A survey on 5g usage scenarios and traffic models. *IEEE Communications Surveys and Tutorials*, 22(2):905–929, 2020.

[7] Shengli Pan, Zhiyong Zhang, Tao Xue, and Guangmin Hu. Enhancing availability for the mec service: Cvar-based computation offloading. In *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 342–347. IEEE, 2020.

[8] Gartner. What edge computing means for infrastructure and operations leaders. https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders, 2018. Online; accessed 12-04-2024.

[9] Gartner Research. Predicts 2024: Edge computing technologies are gaining traction and maturity. Technical Report 4850031, Gartner, October 2023. Published: 18 October 2023.

[10] EdgeIR. Three indications of where edge computing is headed, 2024. Accessed: 2024-03-25.

[11] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. *Mobile Edge Computing: A key technology towards 5G*. ETSI, first edition, September 2015.

[12] ETSI. Multi-access Edge Computing. `https://www.etsi.org/technologies/multi-access-edge-computing`. (Accessed: 2023-11-18).

[13] Dongwook Kim and 3GPP MCC. Edge computing. `https://www.3gpp.org/technologies/edge-computing`, June 2023. Online; accessed 12-04-2024.

[14] European Telecommunications Standards Institute. Welcome to the world of standards! `https://www.etsi.org/`, 2024. Accessed: 2023-11-18.

[15] ETSI. ETSI Multi-access Edge Computing (MEC); Study on MEC Support for V2X Use Cases. Technical report, European Telecommunications Standards Institute, 2022.

[16] ETSI. ETSI TS 122 186 V16.2.0 (2020-11) - Service requirements for enhanced V2X scenarios (3GPP TS 22.186 version 16.2.0 Release 16). Technical report, ETSI, November 2020.

[17] ETSI. GS MEC 003 V2.2.1: Multi-access Edge Computing (MEC); Framework and Reference Architecture, 2020.

[18] ETSI. MEC in 5G networks - White Paper No. 28, Jun 2018.

[19] Hadeel Abdah, João Paulo Barraca, and Rui L. Aguiar. Handover prediction integrated with service migration in 5g systems. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–7, 2020.

[20] Gianfranco Nencioni, Rosario G. Garroppo, and Ruxandra F. Olimid. 5G Multi-Access Edge Computing: A Survey on Security, Dependability, and Performance. *IEEE Access*, 11:63496–63533, 2023.

[21] ETSI MEC: An Introduction (almost) everything you want to know about ETSI MEC. `https://www.etsi.org/images/files/technologies/ETSI-MEC-Public-Overview.pdf`. Accessed: April 12, 2024.

[22] ETSI. ETSI GS MEC 009 V2.2.1 (2020-10) GROUP SPECIFICATION Multi-access Edge Computing (MEC); General principles, patterns and common aspects of MEC Service APIs. Technical Report GS MEC 009, European Telecommunications Standards Institute, October 2020.

[23] ETSI. GS MEC 012 V2.1.1: Multi-access Edge Computing (MEC); Radio Network Information API, Dec 2019.

[24] ETSI. GS MEC 002 V2.1.1: Multi-access Edge Computing (MEC); Phase 2: Use Cases and Requirements, 2018.

[25] ETSI. GS MEC 013 V2.1.1: Multi-access Edge Computing (MEC); Location API, Sep 2019.

[26] ETSI. ETSI GS MEC 014 V2.1.1 (2021-03) GROUP SPECIFICATION Multi-access Edge Computing (MEC); UE Identity API. Group Specification 014, ETSI, March 2021.

[27] ETSI. *ETSI GS MEC 011 V2.2.1: Multi-access Edge Computing (MEC); Edge Platform Application Enablement*, 2022. `https://www.etsi.org/deliver/etsi_gs/MEC/001_099/011/02.02.01_60/gs_MEC011v020201p.pdf`.

[28] ETSI (European Telecommunications Standards Institute). ETSI GS MEC 030 V2.2.1 (2022-05) Multi-access Edge Computing (MEC); V2X Information Service API. Group Specification V2.2.1, ETSI, May 2022.

[29] Ibrahim Afolabi, Tarik Taleb, Konstantinos Samdanis, Adlen Ksentini, and Hannu Flinck. Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. *IEEE Communications Surveys & Tutorials*, 20(3):2429–2453, 2018.

[30] Madhukrishna Priyadarsini and Padmalochan Bera. Software defined networking architecture, traffic management, security, and placement: A survey. *Computer Networks*, 192:108047, 2021.

[31] Mudassar Hussain, Nadir Shah, Rashid Amin, Sultan S Alshamrani, Aziz Alotaibi, and Syed Mohsan Raza. Software-defined networking: Categories, analysis, and future directions. *Sensors*, 22(15):5551, 2022.

[32] Gianfranco Nencioni, Rosario G Garroppo, Andres J Gonzalez, Bjarne E Helvik, and Gregorio Procissi. Orchestration and control in software-defined 5g networks: Research challenges. *Wireless Communications and Mobile Computing*, 2018, 2018.

[33] Pablo Fondo-Ferreiro, Felipe Gil-Castineira, Francisco Javier González-Castaño, and David Candal-Ventureira. A software-defined networking solution for interconnecting network functions in service-based architectures. *IEEE Access*, 10:19905–19916, 2022.

[34] Marian Seliuchenko, Mykola Beshley, Olha Shpur, Nadiia Seliuchenko, Mykhailo Klymash, and Halyna Beshley. Software-defined multi-access edge/cloud computing for 5g/6g time-critical services. In *2023 17th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, volume 1, pages 1–4. IEEE, 2023.

[35] Syed Danial Ali Shah, Mark A Gregory, Shuo Li, Ramon dos Reis Fontes, and Ling Hou. Sdn-based service mobility management in mec-enabled 5g and beyond vehicular networks. *IEEE Internet of Things Journal*, 9(15):13425–13442, 2022.

[36] Juliver Gil Herrera and Juan Felipe Botero. Resource allocation in nfv: A comprehensive survey. *IEEE Transactions on Network and Service Management*, 13(3):518–532, 2016.

[37] European Telecommunications Standards Institute (ETSI). Network Functions Virtualization (NFV). `https://www.etsi.org/images/files/ETSITechnologyLeaflets/NetworkFunctionsVirtualization.pdf`, Accessed: April 12, 2024.

[38] European Telecommunications Standards Institute (ETSI). Network Functions Virtualisation (NFV); Architectural Framework. `https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf`, 2014.

[39] Kiril Antevski, Carlos J. Bernardos, Luca Cominardi, Antonio de la Oliva, and Alain Mourad. On the integration of nfv and mec technologies: architecture analysis and benefits for edge robotics. *Computer Networks*, 175:107274, 2020.

[40] ETSI. GR MEC 017 V1.1.1: Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV environment, 2018.

[41] Navid Nikaein, Mahesh K Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. Openairinterface: A flexible platform for 5g research. *ACM SIGCOMM Computer Communication Review*, 44(5):33–38, 2014.

[42] UERANSIM. `https://github.com/aligungr/UERANSIM`, Last accessed 20 March 2023. GitHub.

[43] Open5GCore. `https://www.open5gcore.org/`, Last accessed 20 March 2023. GitHub.

[44] Open5GS. `https://open5gs.org/open5gs/docs/guide/01-quickstart/`, Last accessed 20 March 2023. GitHub.

[45] Free5GC. https://github.com/free5gc/free5gc, Last accessed 20 March 2023. GitHub.

[46] Giovanni Nardini, Dario Sabella, Giovanni Stea, Purvi Thakkar, and Antonio Virdis. Simu5g–an omnet++ library for end-to-end performance evaluation of 5g networks. *IEEE Access*, 8:181176–181191, 2020.

[47] Intel. OpenNESS Edge Applications. https://github.com/smart-edge-open/edgeapps, Last accessed 20 March 2023. GitHub.

[48] EdgeX. EdgeX Foundry. https://www.edgexfoundry.org/get-started/, Last accessed 20 March 2023.

[49] Roberto Riggio. LightEDGE. https://lightedge.io/, Last accessed 20 March 2023.

[50] Roy Michel, Kevin Di Lallo, and Gazda Robert. AdvantEDGE: A Mobile Edge Emulation Platform (MEEP). https://github.com/InterDigitalInc/AdvantEDGE, Last accessed 20 March 2023. GitHub.

[51] Navid Nikaein, Xenofon Vasilakos, and Anta Huang. Ll-mec: Enabling low latency edge applications. In *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pages 1–7, 2018.

[52] Docker. https://www.docker.com, Last accessed 20 March 2023.

[53] Kubernetes. https://kubernetes.io, 2022. Last accessed 4 October 2022.

[54] OpenInfra Foundation Project. OpenStack. https://www.openstack.org/software/, Last accessed 20 March 2023.

[55] ETSI OSM. Open Source MANO (OSM). https://osm.etsi.org/, Last accessed 20 March 2023.

[56] Open Baton. OpenBaton Documentation. https://openbaton-docs.readthedocs.io/en/3.0.0/, Last accessed 20 March 2023.

[57] LightMANO. The LightMANO Platform. https://github.com/lightmano, Last accessed 20 March 2023. GitHub.

[58] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, January 2011.

[59] SUMO. SUMO V2X Communication Research (Platooning and CIM). https://github.com/sraddon/SUMO-V2X-Communication-Research-Platooning-and-CIM, Last accessed 20 March 2023. GitHub.

[60] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4), 2012.

[61] The Linux Foundation. Prometheus. `https://prometheus.io/`, Last accessed 20 March 2023.

[62] Torkel Ödegaard. . `https://grafana.com/oss/grafana/`. (Accessed: 2023-11-18).

[63] ETSI MEC Sandbox. `https://try-mec.etsi.org/`. Accessed: April 13, 2024.

[64] Robertqiu. Connected Vehicle Blueprint(Aka CVB). `https://wiki.akraino.org/pages/viewpage.action?pageId=9601601`, 2021.

[65] The Eclipse fog05 Community. eclipse-fog05/fog05. `https://github.com/eclipse-fog05/fog05`. Accessed: April 13, 2024.

[66] Edgegallery. GitHub, 2019–2021. (Accessed: 2023-11-18).

[67] EURECOM MEC Platform. `https://gitlab.eurecom.fr/oai/orchestration/blueprints/-/blob/master/mep/README.md`. Accessed: April 13, 2024.

[68] i-MEC. `https://www.italtel.com/products/multi-access-edge-computing-mec-platform/`. Accessed: April 13, 2024.

[69] ccicconetti. ccicconetti/serverlessonedge. `https://github.com/ccicconetti/serverlessonedge`. Accessed: April 13, 2024.

[70] Dario Sabella, Vadim Sukhomlinov, Linh Trang, Sami Kekki, Pietro Paglierani, Ralf Rossbach, Xinhui Li, Yonggang Fang, Dan Druta, Fabio Giust, Luca Cominardi, Walter Featherstone, Bob Pike, and Shlomi Hadad. Developing software for multi-access edge computing. White Paper No. 20, ETSI, February 2019.

[71] Kubernetes documentation. `https://kubernetes.io/docs`. Accessed: April 13, 2024.

[72] Pedro Escaleira, Miguel Mota, Diogo Gomes, João P Barraca, and Rui L Aguiar. Multi-access edge computing as a service. In *2022 18th International Conference on Network and Service Management (CNSM)*, pages 177–183. IEEE, 2022.

[73] Annisa Sarah, Gianfranco Nencioni, and Md. Muhidul I. Khan. Resource allocation in multi-access edge computing for 5g-and-beyond networks. *Computer Networks*, 227:109720, 2023.

[74] Md Muhidul I Khan and Gianfranco Nencioni. Resource allocation in networking and computing systems: a security and dependability perspective. *IEEE Access*, 2023.

[75] Mandar Datar. *Resource allocation and pricing in 5g network slicing*. PhD thesis, Université d'Avignon, 2022.

[76] Hassan Halabian. Distributed resource allocation optimization in 5g virtualized networks. *IEEE Journal on Selected Areas in Communications*, 37(3):627–642, 2019.

[77] Ivan Čilić, Petar Krivić, Ivana Podnar Žarko, and Mario Kušek. Performance evaluation of container orchestration tools in edge computing environments. *Sensors*, 23(8):4008, 2023.

[78] Hao Luo and Hung-Yu Wei. Resource orchestration at the edge: Intelligent management of mmwave ran and gaming application qoe enhancement. *IEEE Transactions on Network and Service Management*, 20(1):385–399, 2022.

[79] Dario Sabella, Andrew Alleman, Ellen Liao, Miltiadis Filippou, Zongrui Ding, Leonardo Gomes Baltar, Srikathyayani Srikanteswara, Krishna Bhuyan, Ozgur Oyman, Gershon Schatzberg, Neal Oliver, Ned Smith, Sharad D. Mishra, Purvi Thakkar, and Samar Shailendra. Edge computing: from standard to actual infrastructure deployment and software development. White Paper, 10 2019. Revised August 2021.

[80] Zeinab Bakhshi, Guillermo Rodriguez-Navas, and Hans Hansson. Fault-tolerant permanent storage for container-based fog architectures. In *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, volume 1, pages 722–729. IEEE, 2021.

[81] ETSI. *ETSI GS MEC 021 V2.1.1 : Multi-access Edge Computing (MEC); Application Mobility Service API*, 2020. `https://www.etsi.org/deliver/etsi_gs/MEC/001_099/021/02.01.01_60/gs_MEC021v020101p.pdf`.

[82] F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante, W. Featherstone, and D. Munaretto. Multi-access edge computing: The driver behind the wheel of 5g-connected cars. *IEEE Communications Standards Magazine*, 2(3):66–73, 2018.

[83] Ngmn alliance selects c-v2x technology for the connected car. https://www.telecomtv.com/content/automotive/ngmn-alliance-selects-c-v2x-technology-for-the-connected-car-31854/, 2018. Online; accessed 12-04-2024.

[84] 5G Americas. Wireless technology evolution towards 5g. 3gpp release 13 and release 15 to beyond, 2017.

[85] Ignacio Llatser, Thomas Michalke, Maxim Dolgov, Florian Wildschütte, and Hendrik Fuchs. Cooperative automated driving use cases for 5g v2x communication. In *2019 IEEE 2nd 5G World Forum (5GWF)*, pages 120–125. IEEE, 2019.

[86] Khabaz Sehla, Thi Mai Trang Nguyen, Guy Pujolle, and Pedro Braconnot Velloso. Resource allocation modes in c-v2x: from lte-v2x to 5g-v2x. *IEEE Internet of Things Journal*, 9(11):8291–8314, 2022.

[87] 3rd Generation Partnership Project (3GPP). 3GPP TS 34.108 V3.1.0 (2000-09) - Technical Specification 3rd Generation Partnership Project; Technical Specification Group Terminals; Common Test Environments for User Equipment (UE) Conformance Testing (Release 1999). Technical report, 3GPP, September 2000.

[88] Athanasios Kanavos, Dimitrios Fragkos, and Alexandros Kaloxylos. V2x communication over cellular networks: Capabilities and challenges. In *Telecom*, volume 2, pages 1–26. MDPI, 2021.

[89] ETSI (European Telecommunications Standards Institute). ETSI GR MEC 022 V2.1.1 (2018-09) - Multi-access Edge Computing (MEC); Study on MEC Support for V2X Use Cases. Technical report, ETSI, September 2018.

[90] ETSI (European Telecommunications Standards Institute). ETSI TR 126 985 V17.0.0 (2022-05) - Technical Report 5G; Vehicle-to-everything (V2X); Media handling and interaction (3GPP TR 26.985 version 17.0.0 Release 17). Technical report, ETSI, May 2022.

[91] Quang-Huy Nguyen, Michel Morold, Klaus David, and Falko Dressler. Car-to-pedestrian communication with mec-support for adaptive safety of vulnerable road users. *Computer Communications*, 150:83–93, 2020.

[92] 5GAA Automotive Association. Cross working group work item gmec4auto moving towards federated mec demos/trials (global mec). Technical report, 5GAA Automotive Association, 2023.

[93] Ali Esmaeily and Katina Kralevska. Small-scale 5g testbeds for network slicing deployment: A systematic review. *Wireless Communications and Mobile Computing*, 2021:1–26, 2021.

[94] Q. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W. Hwang, and Z. Ding. A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE Access*, 8:116974–117017, 2020.

[95] Francesco Spinelli and Vincenzo Mancuso. Toward enabled industrial verticals in 5g: A survey on mec-based approaches to provisioning and flexibility. *IEEE Communications Surveys & Tutorials*, 23(1):596–630, 2020.

[96] P Cruz, N Achir, and AC Viana. On the edge of the deployment: A survey on multi-access edge computing. *ACM Computing Surveys (CSUR)*, 2022.

[97] 5GCity. SDK. `https://github.com/5GCity/5GCity-SDK`, Last accessed 20 March 2023. GitHub repository.

[98] 5GCity. 5GCity Neural Hosting Platform-Installation Instructions. `https://www.5gcity.eu/installation-instructions/`, Last accessed 20 March 2023.

[99] Navid Nikaein, Chia-Yu Chang, and Konstantinos Alexandris. Mosaic5g: Agile and flexible service platforms for 5g research. *ACM SIGCOMM Computer Communication Review*, 48(3):29–34, 2018.

[100] Xenofon Foukas, Navid Nikaein, Mohamed M Kassem, Mahesh K Marina, and Kimon Kontovasilis. Flexran: A flexible and programmable platform for software-defined radio access networks. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, pages 427–441, 2016.

[101] 5GINFIRE. 5GINFIRE-University of Bristol. `https://5ginfire.eu/wp-content/uploads/2018/10/university-of-bristol-testbed-short-description-5ginfire.pdf?x18504`, Last accessed 20 March 2023.

[102] EdgeX. EdgeX Foundry Use Cases. `https://www.edgexfoundry.org/why-edgex/`, Last accessed 20 March 2023.

[103] Priyal Thakkar, Shashvat Sanadhya, Pimmy Gandotra, and Brejesh Lall. A 5g openairinterface (oai) testbed with mec: Deployment, application testing and slicing support. In *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 757–762, 2023.

[104] Madhuri Annavazzala, Arun Kant Dubey, Supriya Dilip Tambe, Bheemarjuna Reddy Tamma, and Antony A Franklin. Demonstration of a v2x use case using mec-assisted 5g emulation framework. In *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 210–212, 2023.

[105] Arzad Alam Kherani, Gaurav Shukla, Shashvat Sanadhya, Neha Vasudev, Muneeb Ahmed, Ashish Singh Patel, Rashi Mehrotra, Brejesh Lall, Huzur Saran, Mythili Vutukuru, Abhishek Singh, Sushila Seshasayee, Vinodh R Viswakumar, and Kishore Loganathan. Development of mec system for indigenous 5g test-bed. In *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 131–133, 2021.

[106] G. Nardini, A. Virdis, G. Stea, and A. Buono. Simulte-mec: Extending simulte for multi-access edge computing. In A. Förster, A. Udugama, A. Virdis, and G. Nardini, editors, *Proceedings of the 5th International OMNeT++ Community Summit*, volume 56 of *EPiC Series in Computing*, pages 35–42. EasyChair, 2018.

[107] Stefan Senk, Hosein K Nazari, How-Hang Liu, Giang T Nguyen, and Frank HP Fitzek. Open-source testbeds for integrating time-sensitive networking with 5g and beyond. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pages 1–7. IEEE, 2023.

[108] H2020-ict-41-2020 vital-5g project. `https://www.vital5g.eu/`, December 2020. Online; accessed 12-04-2024.

[109] K. Trichias, G. Landi, E. Seder, J. Marquez-Barja, R. Frizzell, M. Iordache, and P. Demestichas. Vital-5g: Innovative network applications (netapps) support over 5g connectivity for the transport & logistics vertical. In *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pages 437–442, 2021.

[110] Vincent Charpentier, Nina Slamnik-Kriještorac, Lian Xiangyu, João Francisco Nunes Pinheiro, Cristina Costa, and Johann Marquez-Barja. On enhancing transport & logistics sectors with 5g testbeds and edge network applications (edgeapps). In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6. IEEE, 2023.

[111] Vincent Charpentier, Nina Slamnik-Kriještorac, Juan Brenes, Andreas Gavrielides, Marius Iordache, Georgios Tsiouris, Lian Xiangyu, and Johann M Marquez-Barja. Dynamic and quality-aware network slice management in 5g testbeds. In *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pages 611–616. IEEE, 2023.

[112] Luis Tomas Bolivar, Christos Tselios, Daniel Mellado Area, and George Tsolis. On the deployment of an open-source, 5g-aware evaluation testbed. In *2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pages 51–58. IEEE, 2018.

[113] Ming-Yen Wu, Jiun-Cheng Huang, Yuan-Mao Hung, Cheng-Yi Chien, Jack Shi–Jie Luo, and Shuo-Peng Liang. The edge cloud implementation and application of transnational smart factory of 5g private network. In *2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–6, 2022.

[114] Josef Hammer and Hermann Hellwagner. Efficient transparent access to 5g edge services. In *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*, pages 91–96, 2022.

[115] 5GVINNI. 5GVINNI Munich experimentation facility site. `https://www.5g-vinni.eu/main-facility-sites/`, Last accessed 20 March 2023.

[116] 5GVINNI. 5G-VINNI Solution facility sites High Level Design (HLD) - v1 Munich Facility Site annex. `https://www.5g-vinni.eu/wp-content/uploads/2019/02/5g-vinni_d2.1_annex_a7_munich.pdf`, Last accessed 20 March 2023.

[117] Christos Politis, Konstantinos Liolis, Marius Corici, Eric Troudt, Zsolt Szabó, and Joe Cahill. Design of moving experimentation facility to showcase satellite integration into 5g. In *2019 European Conference on Networks and Communications (EuCNC)*, pages 177–181, 2019.

[118] Marius Corici, Konstantinos Liolis, Frank Burkhardt, Ilie Gheorghe-Pop, Stefan Covaci, Christos Politis, Alexander Geurtz, Jan Koernicke, Florian Völk, and Adam Kapovits. Satis5: A 5g testbed integrating satellite and terrestrial infrastructures. In *Advanced Satellite Multimedia Systems Conference (ASMS), Berlin, Germany*, 2018.

[119] University of Antwerpen IDLab. Smart Highway. `https://www.uantwerpen.be/en/research-groups/idlab/infrastructure/smart-highway/`, Last accessed 20 March 2023.

[120] 5GCarmen. 5G-CARMEN Connected and Automated Road Mobility in the European union. `https://5g-ppp.eu/wp-content/uploads/2020/06/5G-Carmen.pdf`, 2020.

[121] Matti Kutila, Kimmo Kauvo, Pasi Pyykönen, Xiaoyun Zhang, Victor Garrido Martinez, Yinxiang Zheng, and Shen Xu. A c-v2x/5g field study for supporting automated driving. In *32nd IEEE Intelligent Vehicles Symposium, IV'21*, pages 315–320. IEEE Institute of Electrical and Electronic Engineers, 2021.

[122] Matti Kutila, Lasse Nykänen, and Matti Lankinen. 5g-drive: Eu china c-v2x collaboration. In *8th Transport Research Arena, TRA 2020-Conference cancelled*, page 147. Liikenne-ja viestintävirasto Traficom, 2020.

[123] M Jalal Khan, Manzoor Ahmed Khan, Obaid Ullah, Sumbal Malik, Farkhund Iqbal, Hesham El-Sayed, and Sherzod Turaev. Augmenting ccam infrastructure for creating smart roads and enabling autonomous driving. *Remote Sensing*, 15(4):922, 2023.

[124] Nina Slaninik-Krijestorac, Mauro Femminella, Girma M Yilma, Marco Liebsch, Gianluca Reali, and Johann M Marquez-Barja. Orchestrating distributed 5g edges for automotive cross-border trials: Validation of an experimental prototype. *ITU journal on future and evolving technologies*, 4(2):306–324, 2023.

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

[126] Ridha Soua, Ion Turcanu, Florian Adamsky, Detlef Führer, and Thomas Engel. Multi-access edge computing for vehicular networks: A position paper. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2018.

[127] Thomas Soenen, Wouter Tavernier, Manuel Peuster, Felipe Vicens, George Xilouris, Stavros Kolometsos, Michail-Alexandros Kourtis, and Didier Colle. Empowering network service developers: enhanced nfv devops and programmable mano. *IEEE Communications Magazine*, 57(5):89–95, 2019.

[128] Nathan F Saraiva De Sousa, Danny A Lachos Perez, Raphael V Rosa, Mateus AS Santos, and Christian Esteve Rothenberg. Network service orchestration: A survey. *Computer Communications*, 142:69–94, 2019.

[129] Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J García Clemente, and Gregorio Martínez Pérez. Automatic monitoring management for 5g mobile networks. *Procedia Computer Science*, 110:328–335, 2017.

[130] Oleksandr Zhdanenko, Jianhui Liu, Roberto Torre, Stanislav Mudriievskiy, Hani Salah, Giang T Nguyen, and HP Frank Fitzek. Demonstration of mobile edge cloud for 5g connected cars. In *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–2. IEEE, 2019.

[131] Manuel Peuster, Michael Marchetti, Gerardo García de Blas, and Holger Karl. Automated testing of nfv orchestrators against carrier-grade multi-pop scenarios using emulation-based smoke testing. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):172, 2019.

[132] Girma M Yilma, Faqir Zarrar Yousaf, Vincenzo Sciancalepore, and Xavier Costa-Perez. On the challenges and kpis for benchmarking open-source nfv mano systems: Osm vs onap. *arXiv preprint arXiv:1904.10697*, 2019.

[133] Jianqi Liu, Jiafu Wan, Bi Zeng, Qinruo Wang, Houbing Song, and Meikang Qiu. A scalable and quick-response software defined vehicular network assisted by mobile edge computing. *IEEE Communications Magazine*, 55(7):94–100, 2017.

[134] Álvaro Santos, Jorge Bernardino, and Noélia Correia. Automated application deployment on multi-access edge computing: A survey. *IEEE Access*, 2023.

[135] Vojdan Kjorveziroski, Cristina Bernad Canto, Katja Gilly, and Sonja Filiposka. Implementing multi-access edge computing with kubernetes, 2022. Publisher: ICT Innovations.

[136] Harrison Mfula, Antti Ylä-Jääski, and Jukka K Nurminen. Seamless kubernetes cluster management in multi-cloud and edge 5g applications. In *International Conference on High Performance Computing & Simulation*, 2021.

[137] Nina Slamnik-Kriještorac, Michael Peeters, Steven Latré, and Johann M. Marquez-Barja. Analyzing the impact of vim systems over the mec management and orchestration in vehicular communications. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6, 2020.

[138] Sergio Barrachina, Miquel Payaró, Panagiotis Kokkinos, Polyzois Soumplis, Konstantinos Kontodimas, Emmanouel Varvarigos, John Vardakas, Roberto González, Giuseppe Siracusano, Davide Sanvito, and Roberto Bifulco. Deliverable d4.2: Initial report on elastic mec platform design and data-driven orchestration and automation. Deliverable D4.2, MARSAL Project, 6 2022. Grant Agreement No. 101017171, Acronym: MARSAL, Full Title: Machine learning-based, networking and computing infrastructure resource management of 5G and beyond intelligent networks.

[139] Simone Bolettieri, Dinh Thai Bui, and Raffaele Bruno. Towards end-to-end application slicing in multi-access edge computing systems: Architecture discussion and proof-of-concept. *Future Generation Computer Systems*, 136:110–127, 2022.

[140] Jude Okwuibe, Juuso Haavisto, Erkki Harjula, Ijaz Ahmad, and Mika Yliant-tila. Sdn enhanced resource orchestration of containerized edge applications for industrial iot. *IEEE Access*, 8:229117–229131, 2020.

[141] Daniil Ermolenko, Claudia Kilicheva, Ammar Muthanna, and Abdukodir Khakimov. Internet of things services orchestration framework based on kubernetes and edge computing. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pages 12–17. IEEE, 2021.

[142] I Labriji, F Meneghello, D Cecchinato, S Sesia, E Perraud, EC Strinati, and M Rossi. Mobility aware and dynamic migration of mec services for the internet of vehicles. *IEEE Transactions on Network and Service Management*, 18(1):570–584, Jan 2021.

[143] T Kondo, K Isawaki, and K Maeda. Development and evaluation of the mec platform supporting the edge instance mobility. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pages 193–198. IEEE, Jul 2018.

[144] J Okwuibe, J Haavisto, E Harjula, I Ahmad, and M Ylianttila. Orchestrating service migration for low power mec-enabled iot devices. *arXiv preprint arXiv:1905.12959*, May 2019.

[145] Intel. *Intel® Smart Edge Open*, 2022. Last accessed 4 October 2022.

[146] Pablo Fondo-Ferreiro, Alberto Estévez-Caldas, Rubén Pérez-Vaz, Felipe Gil-Castiñeira, Francisco Javier González-Castaño, Santiago Rodríguez-García, Xosé Ramón Sousa-Vázquez, Diego López, and Carmen Guerrero. Seamless multi-access edge computing application handover experiments. In *2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*, pages 1–6, 2021.

[147] Kiranpreet Kaur, Fabrice Guillemin, and Francoise Sailhan. Container placement and migration strategies for cloud, fog, and edge data centers: A survey. *International Journal of Network Management*, 32(6):e2212, 2022.

[148] J Schrettenbrunner. Migrating pods in kubernetes. `https://github.com/SSU-DCN/podmigration-operator/blob/main/init-cluster-containerd-CRIU.md`, 2020. Last accessed 4 October 2022.

[149] SSU-DCN. podmigration-operator. `https://github.com/SSU-DCN/podmigration-operator/blob/main/init-cluster-containerd-CRIU.md`, 2022. Last accessed 4 October 2022.

[150] Paulo Souza Junior, Daniele Miorandi, and Guillaume Pierre. Good shepherds care for their cattle: Seamless pod migration in geo-distributed kubernetes. In *2022 IEEE 6th international conference on fog and edge computing (ICFEC)*, pages 26–33. IEEE, 2022.

[151] SeungYong Oh and JongWon Kim. Stateful container migration employing checkpoint-based restoration for orchestrated container clusters. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 25–30. IEEE, 2018.

[152] Paulo Ricardo Rodrigues de Souza Junior. *Stateful application migration in geo-distributed systems.* PhD thesis, Université de Rennes, 2022.

[153] Daniel Fraunholz, Richard Schörghofer-Vrinssen, Hartmut König, Wolfgang Mühlbauer, and Richard Zahoranksy. Mobility-enabling edge cloud infrastructure: Testbed and experimental evaluation. In *2021 IEEE Cloud Summit (Cloud Summit)*, pages 19–24, 2021.

[154] Qi Liu, Ruichao Mo, Xiaolong Xu, and Xu Ma. Multi-objective resource allocation in mobile edge computing using paes for internet of things. *Wireless networks*, pages 1–13, 2020.

[155] Zan Li, Zhongling Zhao, Jia Shi, Jiangbo Si, Pei Xiao, Rahim Tafazolli, and Hang Hu. Multi-objective deep reinforcement learning assisted resource allocation for mec-caching-coexist system. *IEEE internet of things journal*, 2023.

[156] Peng Wang, Kenli Li, Bin Xiao, and Keqin Li. Multiobjective optimization for joint task offloading, power assignment, and resource allocation in mobile edge computing. *IEEE internet of things journal*, 9(14):11737–11748, 2021.

[157] Taha Alfakih, Mohammad Mehedi Hassan, and Muna Al-Razgan. Multi-objective accelerated particle swarm optimization with dynamic programing technique for resource allocation in mobile edge computing. *IEEE Access*, 9:167503–167520, 2021.

[158] Yu Bi, Carlos Colman Meixner, Monchai Bunyakitanon, Xenofon Vasilakos, Reza Nejabati, and Dimitra Simeonidou. Multi-objective deep reinforcement learning assisted service function chains placement. *IEEE Transactions on Network and Service Management*, 18(4):4134–4150, 2021.

[159] Liqing Liu, Xiaoming Yuan, Decheng Chen, Ning Zhang, Haifeng Sun, and Amir Taherkordi. Multi-user dynamic computation offloading and resource allocation in 5g mec heterogeneous networks with static and dynamic subchannels. *IEEE Transactions on Vehicular Technology*, 2023.

[160] Chubo Liu, Fan Tang, Yikun Hu, Kenli Li, Zhuo Tang, and Keqin Li. Distributed task migration optimization in mec by extending multi-agent deep reinforcement learning approach. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1603–1614, 2020.

[161] Muhammad Rizwan Anwar, Shangguang Wang, Muhammad Faisal Akram, Salman Raza, and Shahid Mahmood. 5g-enabled mec: A distributed traffic steering for seamless service migration of internet of vehicles. *IEEE Internet of Things Journal*, 9(1):648–661, 2022.

[162] Kun Cao, Liying Li, Yangguang Cui, Tongquan Wei, and Shiyan Hu. Exploring placement of heterogeneous edge servers for response time minimization in mobile edge-cloud computing. *IEEE Transactions on Industrial Informatics*, 17(1):494–503, 2021.

[163] Tiago Koketsu Rodrigues, Katsuya Suto, and Nei Kato. Edge cloud server deployment with transmission power control through machine learning for 6g internet of things. *IEEE Transactions on Emerging Topics in Computing*, 9(4):2099–2108, 2021.

[164] Syed Danial Ali Shah, Mark A Gregory, Shuo Li, and Ramon Dos Reis Fontes. Sdn enhanced multi-access edge computing (mec) for e2e mobility and qos management. *IEEE Access*, 8:77459–77469, 2020.

[165] Pablo Fondo-Ferreiro, David Candal-Ventureira, Felipe Gil-Castiñeira, Francisco Javier González-Castaño, and Diarmuid Collins. Experimental evaluation of end-to-end flow latency reduction in softwarized cellular networks through dynamic multi-access edge computing. In *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1310–1315, 2021.

[166] Tung V. Doan, Giang T. Nguyen, Martin Reisslein, and Frank H. P. Fitzek. Fast: Flexible and low-latency state transfer in mobile edge computing. *IEEE Access*, 9:115315–115334, 2021.

[167] Ioannis Sarrigiannis, Elli Kartsakli, Kostas Ramantas, Angelos Antonopoulos, and Christos Verikoukis. Application and network vnf migration in a mec-enabled 5g architecture. In *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6, 2018.

[168] Constantine Ayimba, Michele Segata, Paolo Casari, and Vincenzo Mancuso. Driving under influence: Robust controller migration for mec-enabled platooning. *Computer Communications*, 194:135–147, 2022.

[169] Alcardo Alex Barakabitze, Arslan Ahmad, Rashid Mijumbi, and Andrew Hines. 5g network slicing using sdn and nfv: A survey of taxonomy, architectures and future challenges. *Computer Networks*, 167:106984, 2020.

[170] Shuo Wang, Tao Sun, Hongwei Yang, Xiaodong Duan, and Lu Lu. 6g network: Towards a distributed and autonomous system. In *2020 2nd 6G wireless summit (6G SUMMIT)*, pages 1–5. IEEE, 2020.

[171] Qian Chen, Zheng Guo, Weixiao Meng, Shuai Han, Cheng Li, and Tony Q. S. Quek. A survey on resource management in joint communication and computing-embedded sagin. *IEEE Communications Surveys & Tutorials*, pages 1–1, 2024.

[172] David Witkowski. *Bridging the Gap - 21st Century Wireless Telecommunications Handbook (2nd Edition, 2019)*. Joint Venture Silicon Valley Report, 12 2019.

[173] Guangxu Zhu, Dongzhu Liu, Yuqing Du, Changsheng You, Jun Zhang, and Kaibin Huang. Toward an intelligent edge: Wireless communication meets machine learning. *IEEE Communications Magazine*, 58(1):19–25, 2020.

[174] 5G Automotive Association (5GAA). Toward fully connected vehicles: Edge computing for advanced automotive communications. *5GAA Reports*, December 2017.

[175] ETSI. Gs mec 030 v2.2.1: Multi-access edge computing (mec); v2x information service api, 2022.

[176] Hanwen Cao, Sandip Gangakhedkar, Ali Ramadan Ali, Mohamed Gharba, and Joseph Eichinger. A testbed for experimenting 5g-v2x requiring ultra reliability and low-latency. In *WSA 2017; 21th International ITG Workshop on Smart Antennas*, pages 1–4. VDE, 2017.

[177] Khadige Abboud, Hassan Aboubakr Omar, and Weihua Zhuang. Interworking of dsrc and cellular network technologies for v2x communications: A survey. *IEEE transactions on vehicular technology*, 65(12):9457–9470, 2016.

[178] Fabio Giust, Vincenzo Sciancalepore, Dario Sabella, Miltiades C Filippou, Simone Mangiante, Walter Featherstone, and Daniele Munaretto. Multi-access edge computing: The driver behind the wheel of 5g-connected cars. *IEEE Communications Standards Magazine*, 2(3):66–73, 2018.

[179] Quoc-Viet Pham, Fang Fang, Vu Nguyen Ha, Md Jalil Piran, Mai Le, Long Bao Le, Won-Joo Hwang, and Zhiguo Ding. A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE access*, 8:116974–117017, 2020.

[180] Ling Hou, Mark A Gregory, and Shuo Li. A survey of multi-access edge computing and vehicular networking. *IEEE Access*, 10:123436–123451, 2022.

[181] Francesco Spinelli and Vincenzo Mancuso. Toward enabled industrial verticals in 5g: A survey on mec-based approaches to provisioning and flexibility. *IEEE Communications Surveys and Tutorials*, 23(1):596–630, 2020.

[182] Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, and Dario Sabella. On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys and Tutorials*, 19(3):1657–1681, 2017.

[183] 3GPP. 5g system overview. `https://www.3gpp.org/technologies/5g-system-overview`.

[184] M Series. Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond. *Recommendation ITU*, 2083(0), 2015.

[185] C. R. Storck and F. Duarte-Figueiredo. A survey of 5g technology evolution, standards, and infrastructure associated with vehicle-to-everything communications by internet of vehicles. *IEEE Access*, 8:117593–117614, 2020.

[186] Abderrahime Filali, Amine Abouaomar, Soumaya Cherkaoui, Abdellatif Kobbane, and Mohsen Guizani. Multi-access edge computing: A survey. *IEEE Access*, 8:197017–197046, 2020.

[187] Sami Kekki, Walter Featherstone, Yonggang Fang, Pekka Kuure, Alice Li, Anurag Ranjan, Debashish Purkayastha, Feng Jiangping, Danny Frydman, Gianluca Verin, et al. Mec in 5g networks. *ETSI white paper*, 28(2018):1–28, 2018.

[188] Mehmet Ersue. Etsi nfv management and orchestration-an overview. *Presentation at the IETF*, 88, 2013.

[189] Dario Sabella. *MEC in Virtualized Environments*, pages 159–199. Springer International Publishing, Cham, 2021.

[190] Shanzhi Chen, Jinling Hu, Yan Shi, Ying Peng, Jiayi Fang, Rui Zhao, and Li Zhao. Vehicle-to-everything (v2x) services supported by lte-based systems and 5g. *IEEE Communications Standards Magazine*, 1(2):70–76, 2017.

[191] Abdelkader Aissioui, Adlen Ksentini, Abdelhak Mourad Gueroui, and Tarik Taleb. On enabling 5g automotive systems using follow me edge-cloud concept. *IEEE Transactions on Vehicular Technology*, 67(6):5302–5316, 2018.

[192] Carlos Colman-Meixner, Hamzeh Khalili, Konstantinos Antoniou, Muhammad Shuaib Siddiqui, Apostolos Papageorgiou, Antonino Albanese, Paolo Cruschelli, Gino Carrozzo, Luca Vignaroli, Alexandre Ulisses, et al. Deploying a novel 5g-enabled architecture on city infrastructure for ultra-high definition and immersive media production and broadcasting. *IEEE Transactions on Broadcasting*, 65(2):392–403, 2019.

[193] Sokratis Barmpounakis, George Tsiatsios, Michael Papadakis, Evangelos Mitsianis, Nikolaos Koursioumpas, and Nancy Alonistioti. Collision avoidance in 5g using mec and nfv: The vulnerable road user safety use case. *Computer Networks*, 172:107150, 2020.

[194] Hanwen Cao, Sandip Gangakhedkar, Ali Ramadan Ali, Mohamed Gharba, and Josef Eichinger. A 5g v2x testbed for cooperative automated driving. In *2016 IEEE Vehicular Networking Conference (VNC)*, pages 1–4, 2016.

[195] Mohamed Gharba, Hanwen Cao, Sandip Gangakhedkar, Josef Eichinger, Ali Ramadan Ali, Karthikeyan Ganesan, Varun Jain, Stephan Lapoehn, Tobias Frankiewicz, Tobias Hesse, et al. 5g enabled cooperative collision avoidance: System design and field test. In *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6. IEEE, 2017.

[196] Konstantinos Liolis, Joe Cahill, Eddy Higgins, Marius Corici, Eric Troudt, and Paul Sutton. Over-the-air demonstration of satellite integration with 5g core network and multi-access edge computing use case. In *2019 IEEE 2nd 5G World Forum (5GWF)*, pages 1–5. IEEE, 2019.

[197] FED4FIRE. Citylab. `https://www.fed4fire.eu/testbeds/citylab/`, Last accessed 20 March 2023.

[198] Johann Marquez-Barja, Bart Lannoo, Bart Braem, Carlos Donato, Vasilis Maglogiannis, Siegfried Mercelis, Raf Berkvens, Peter Hellinckx, Maarten Weyn, Ingrid Moerman, and Steven Latré. Smart highway: Its-g5 and c-v2x based testbed for vehicular communications in real environments enhanced by edge/cloud technologies. *manuscript*, 06 2019.

[199] Nina Slamnik-Kriještorac and Johann M Marquez-Barja. Unraveling edge-based in-vehicle infotainment using the smart highway testbed. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–4. IEEE, 2021.

[200] Nina Slamnik-Krijestorac, Girma M Yilma, Marco Liebsch, Faqir Zarrar Yousaf, and Johann Marquez-Barja. Collaborative orchestration of multi-domain edges from a connected, cooperative and automated mobility (ccam) perspective. *IEEE Transactions on Mobile Computing*, 2021.

[201] Jiayue Liang, Fang Liu, Shen Li, and Zhenhua Cai. A comparative research on open source edge computing systems. In *Artificial Intelligence and Security: 5th International Conference, ICAIS 2019, New York, NY, USA, July 26-28, 2019, Proceedings, Part II 5*, pages 157–170. Springer, 2019.

[202] Ioannis P Chochliouros, Anastasia S Spiliopoulou, Alexandros Kostopoulos, Eirini Vasilaki, Uwe Herzog, Athanassios Dardamanis, Tao Chen, Latif Ladid, and Marinos Agapiou. Testbeds for the implementation of 5g in the european union: The innovative case of the 5g-drive project. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 78–92. Springer, 2019.

[203] Mosaic5G. `https://gitlab.eurecom.fr/mosaic5g`, Last accessed 20 March 2023. GitLAB.

[204] 5GINFIRE. https://5ginfire.eu/it-av-automotive-testbed/, Last accessed 20 March 2023.

[205] 5GVINNI. 5GVINNI moving experimentation facility site. https://www.5g-vinni.eu/main-facility-sites/, Last accessed 20 March 2023.

[206] FED4FIRE. Smart highway V2X testbed. https://www.fed4fire.eu/testbeds/smart-highway/, Last accessed 20 March 2023.

[207] 5G-DRIVE. D4.1: V2X Development and Test Plan. https://5g-drive.eu/resources-and-results/project-deliverables/#1552568196209-3ef4b649-a787, Last accessed 20 March 2023. Project Deliverable.

[208] 5GCity. Use cases-5GCity. https://www.5gcity.eu/use-cases/, Last accessed 20 March 2023. Project Deliverable.

[209] Mosaic5G. Use cases-Mosaic5G. https://mosaic5g.io/#portfolioModal1, Last accessed 20 March 2023.

[210] 5GINFIRE. Smart Internet Lab, University of Bristol. https://5ginfire.eu/wp-content/uploads/2018/05/bristol-testbed.pdf, Last accessed 20 March 2023. Project Deliverable.

[211] 5GVINNI. Deliverable D3.1 Specification of services delivered by each of the 5G-VINNI facilities. https://www.5g-vinni.eu/deliverables/, Last accessed 20 March 2023. Project Deliverable.

[212] 5GCarmen. Use cases-5GCarmen . https://5gcarmen.eu/use-cases/, Last accessed 20 March 2023.

[213] MagmaEPC. https://github.com/magma/magma, Last accessed 20 March 2023. GitHub.

[214] NextEPC. https://github.com/nextepc/nextepc, Last accessed 20 March 2023. GitHub.

[215] Gergely Pongrácz, László Molnár, and Zoltán Lajos Kis. Removing roadblocks from sdn: Openflow software switch performance on intel dpdk. In *2013 Second European Workshop on Software Defined Networks*, pages 62–67. IEEE, 2013.

[216] Rogério Leão Santos De Oliveira, Christiane Marie Schweitzer, Ailton Akira Shinoda, and Ligia Rodrigues Prete. Using mininet for emulation and prototyping software-defined networks. In *2014 IEEE Colombian conference on communications and computing (COLCOM)*, pages 1–6. Ieee, 2014.

[217] Linux Foundation Collaborative Project. OpenvSwitch. https://www.openvswitch.org/, Last accessed 20 March 2023.

[218] Syed Danial Ali Shah, Mark A Gregory, and Shuo Li. Cloud-native network slicing using software defined networking based multi-access edge computing: A survey. *IEEE Access*, 9:10903–10924, 2021.

[219] Liehuang Zhu, Md M Karim, Kashif Sharif, Chang Xu, Fan Li, Xiaojiang Du, and Mohsen Guizani. Sdn controllers: A comprehensive analysis and performance evaluation study. *ACM Computing Surveys (CSUR)*, 53(6):1–40, 2020.

[220] OpenDaylight Project. OpenDaylight. https://docs.opendaylight.org/projects/controller/en/latest/dev-guide.html, Last accessed 20 March 2023.

[221] Open Networking Foundation. Open Network Operating System (ONOS). https://wiki.onosproject.org/display/ONOS/ONOS+%3A+An+Overview, Last accessed 20 March 2023.

[222] Stefano Secci, Alessio Diamanti, José Manuel Sanchez Vilchez, Mamoudou Tahirou Bah, Pedra Vizzarreta, Carmen Mas Machuca, Sandra Scott-Hayward, and David Smith. *Security and performance comparison of ONOS and ODL controllers*. PhD thesis, ONOS, 2019.

[223] Intel. OpenNESS. https://github.com/smart-edge-open, Last accessed 20 March 2023. GitHub.

[224] Roberto Riggio. LightEDGE. https://github.com/lightedge/lightedge.github.io/wiki, Last accessed 20 March 2023. GitHub.

[225] Red Hat OpenShift. KVM Hypervisor. https://www.linux-kvm.org/page/Main_Page, Last accessed 20 March 2023.

[226] LXC - Linux Containers. https://github.com/lxc/lxc, Last accessed 20 March 2023. GitHub.

[227] Kostas Katsalis, Navid Nikaein, and Anta Huang. Jox: An event-driven orchestrator for 5g network slicing. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2018.

[228] Tejas Subramanya, Giovanni Baggio, and Roberto Riggio. lightmec: A vendor-agnostic platform for multi-access edge computing. In *2018 14th International Conference on Network and Service Management (CNSM)*, pages 198–204, 2018.

[229] German Aerospace Center (DLR) and others. SUMO User Documentation. https://sumo.dlr.de/docs/index.html, Last accessed 20 March 2023.

[230] Panagiotis Trakadas, Xavi Masip-Bruin, Federico M Facca, Sotirios T Spantideas, Anastasios E Giannopoulos, Nikolaos C Kapsalis, Rui Martins, Enrica Bosani, Joan Ramon, Raül González Prats, et al. A reference architecture for cloud–edge meta-operating systems enabling cross-domain, data-intensive, ml-assisted applications: Architectural overview and key concepts. *Sensors*, 22(22):9003, 2022.

[231] Federico Alvarez, David Breitgand, David Griffin, Pasquale Andriani, Stamatia Rizou, Nikolaos Zioulis, Francesca Moscatelli, Javier Serrano, Madeleine Keltsch, Panagiotis Trakadas, et al. An edge-to-cloud virtualized multimedia service platform for 5g networks. *IEEE Transactions on Broadcasting*, 65(2):369–380, 2019.

[232] Stamatia Rizou, Panagiotis Athanasoulis, Pasquale Andriani, Francesco Iadanza, Panagiotis Trakadas, David Griffin, Morteza Kheirkhah, David Breitgand, Avi Weit, Refik Fatih Ustok, et al. Programmable edge-to-cloud virtualization for 5g media industry: The 5g-media approach. In *Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops: MHDW 2020 and 5G-PINE 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings 16*, pages 95–104. Springer, 2020.

[233] Atakan Aral and Vincenzo De Maio. Simulators and emulators for edge computing. In *Edge Computing: Models, Technologies, and Applications*, chapter 14. Institution of Engineering and Technology (IET), June 2020.

[234] R. Gazda, M. Roy, J. Blakley, A. Sakr, and R. Schuster. Towards open and cross domain edge emulation - the advantedge platform. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 339–344. IEEE, 2021.

[235] M. Symeonides, Z. Georgiou, D. Trihinas, G. Pallis, and M.D. Dikaiakos. Fogify: A fog computing emulation framework. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 42–54. IEEE, 2020.

[236] Advantedge on github. `https://github.com/InterDigitalInc/AdvantEDGE`. (Accessed: 2022-06-20).

[237] J. Burbano, A. Sakr, and R. Schuster. Sliding-window approach for improving response time of mission-critical applications. In *2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, pages 1–7. IEEE, 2020.

[238] J.R. Blakley, R. Iyengar, and M. Roy. Simulating edge computing environments to optimize application experience, 2020. Technical report, available at `http://reports-archive.adm.cs.cmu.edu/ano`.

[239] Video lan website. https://www.videolan.org/index.it.html. (Accessed: 2022-01-31).

[240] Iso/iec 13818-1. https://www.iso.org/standard/75928.html. (Accessed: 2021-05-07).

[241] T. Odegaard. Grafana. https://grafana.com/oss/grafana/. (Accessed: 2021-03-15).

[242] Methods for objective and subjective assessment of speech and video quality - p.808 (06/21). https://www.itu.int/rec/T-REC-P.808-202106-I/en. (Accessed: 2021-08-20).

[243] Quoc-Viet Pham, Fang Fang, Vu Nguyen Ha, Mohammad Javad Piran, Minh Le, Long Bao Le, Won-Joo Hwang, and Zhiguo Ding. A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE Access*, 8:116974–7017, Jun 2020.

[244] Mec ecosystem. https://mecwiki.etsi.org/index.php?title=MEC_Ecosystem, 2022. Last accessed 4 Oct 2022.

[245] JR Blakley, R Iyengar, and M Roy. Simulating edge computing environments to optimize application experience. Technical Report CMU-CS-20-135, School of Computer Science Carnegie Mellon University, Nov 2020.

[246] R Gazda, M Roy, J Blakley, A Sakr, and R Schuster. Towards open and cross domain edge emulation–the advantedge platform. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 339–344. IEEE, Dec 2021.

[247] M Abdulmaksoud, N Dehadrai, J Castrillón, A Sakr, and R Schuster. Edge diagnostics platform: Orchestration and diagnosis model for edge computing infrastructure. In *2021 IEEE International Conference on Edge Computing (EDGE)*, pages 51–59. IEEE, Sep 2021.

[248] J Burbano, A Sakr, and R Schuster. Sliding-window approach for improving response time of mission-critical applications. In *2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, pages 1–7. IEEE, Sep 2020.

[249] A Sakr, S Mohiyadeen, B Vruksharaj, and R Schuster. Qos-aware score-based edge resource allocation model. In *2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, pages 1–7. IEEE, Sep 2020.

[250] Influxdb. https://www.influxdata.com, 2022. Last accessed 4 Oct 2022.

[251] Google kubernetes engine. https://cloud.google.com/kubernetes-engine, 2022. Last accessed 4 October 2022.

[252] InterDigitalInc. Advantedge. https://interdigitalinc.github.io/AdvantEDGE/docs/overview/edge-services/ams/, 2022. Last accessed 4 Oct 2022.

[253] ETSI. *ETSI GS MEC 028 V2.2.1: Multi-access Edge Computing (MEC); WLAN Access Information API*, 2022. https://www.etsi.org/deliver/etsi_gs/MEC/001_099/028/02.02.01_60/gs_MEC028v020201p.pdf.

[254] NGMN. 5G white paper. https://ngmn.org/wp-content/uploads/NGMN_5G_White_Paper_V1_0.pdf, 2015. (Accessed: 2023-11-18).

[255] NGMN. 6G Drivers and Vision V.1.0. https://www.ngmn.org/wp-content/uploads/NGMN-6G-Drivers-and-Vision-V1.0_final.pdf, 2021. (Accessed: 2023-11-18).

[256] 3GPP. 3gpp ts 22.261 v17.9.0, "3rd generation partnership project; technical specification group services and system aspects; service requirements for the 5g system; stage 1 (release 17). *3GPP Technical Specifications*, 2021 December.

[257] Kiryong Ha, Yoshihisa Abe, Zhuo Chen, Wenlu Hu, Brandon Amos, Padmanabhan Pillai, and Mahadev Satyanarayanan. Adaptive vm handoff across cloudlets. *Technical Report CMU-CS-15-113*, 2015.

[258] Kiryong Ha, Yoshihisa Abe, Thomas Eiszler, Zhuo Chen, Wenlu Hu, Brandon Amos, Rohit Upadhyaya, Padmanabhan Pillai, and Mahadev Satyanarayanan. You can teach elephants to dance: Agile vm handoff for edge computing. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, New York, NY, USA, 2017. Association for Computing Machinery.

[259] Mohammed A. Hathibelagal, Rosario G. Garroppo, and Gianfranco Nencioni. Experimental comparison of migration strategies for mec-assisted 5g-v2x applications. *Computer Communications*, 197:1–11, 2023.

[260] Shiqiang Wang, Rahul Urgaonkar, Murtaza Zafer, Ting He, Kevin Chan, and Kin K. Leung. Dynamic service migration in mobile edge-clouds. In *2015 IFIP Networking Conference (IFIP Networking)*, pages 1–9, 2015.

[261] R. G. Garroppo, S. Giordano, and L. Tavanti. A survey on multi-constrained optimal path computation: Exact and approximate algorithms. *Computer Networks*, 54(17):3081–3107, 2010.

[262] Pedro Maristany de las Casas, Antonio Sedeño-Noda, and Ralf Borndörfer. An improved multiobjective shortest path algorithm. *Computers & Operations Research*, page 105424, 2021.

[263] Pedro Maristany de las Casas, Luitgard Kraus, Antonio Sedeño-Noda, and Ralf Borndörfer. Targeted multiobjective dijkstra algorithm, 2021.

[264] Temirlan Kurbanov, Marek Cuchý, and Jiří Vokřínek. Fast one-to-many multicriteria shortest path search. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–10, 2023.

[265] Vajiheh Farhadi, Fidan Mehmeti, Ting He, Thomas F. La Porta, Hana Khamfroush, Shiqiang Wang, Kevin S. Chan, and Konstantinos Poularakis. Service placement and request scheduling for data-intensive applications in edge clouds. *IEEE/ACM Transactions on Networking*, 29(2):779–792, 2021.

[266] Zhenjiang Li and J. J. Garcia-Luna-Aceves. A distributed approach for multi-constrained path selection and routing optimization. In *Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, QShine '06, page 36–es, New York, NY, USA, 2006. Association for Computing Machinery.

[267] ETSI. ETSI GS MEC 003 V2.2.1. Technical report, ETSI, December 2020. (Accessed: 2023-11-18).

[268] ISO. ISO/IEC 13818-1. `https://www.iso.org/standard/75928.html`. (Accessed: 2023-11-18).

[269] ITU-T. Methods for objective and subjective assessment of speech and video quality - P.808 (06/21). `https://www.itu.int/rec/T-REC-P.808-202106-I/en`. (Accessed: 2023-11-18).

[270] Wireshark. Wireshark website. `https://www.wireshark.org/`. (Accessed: 2023-11-18).

[271] Tomasz W. Nowak, Mariusz Sepczuk, Zbigniew Kotulski, Wojciech Niewolski, Rafal Artych, Krzysztof Bocianiak, Tomasz Osko, and Jean-Philippe Wary. Verticals in 5g mec-use cases and security challenges. *IEEE Access*, 9:87251–87298, 2021.

[272] Jan Plachy, Zdenek Becvar, and Emilio Calvanese Strinati. Dynamic resource allocation exploiting mobility prediction in mobile edge computing. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, 2016.

[273] Nafiseh Sharghivand, Lena Mashayekhy, Weibin Ma, and Schahram Dustdar. Time-constrained service handoff for mobile edge computing in 5g. *IEEE Transactions on Services Computing*, 16(3):2241–2253, 2023.

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

[275] Prachi Vinod Wadatkar, Rosario G. Garroppo, and Gianfranco Nencioni. MEC Application Migration by Using AdvantEDGE. In Shui Yu, Bruce Gu, Youyang Qu, and Xiaodong Wang, editors, *Tools for Design, Implementation and Verification of Emerging Information Technologies*, pages 104–118. Springer Nature Switzerland, 2023.

[276] Claudia Campolo, Antonio Iera, Antonella Molinaro, and Giuseppe Ruggeri. MEC support for 5G-V2X use cases through docker containers. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2019.

[277] Francesco Barbarulo, Carlo Puliafito, Antonio Virdis, and Enzo Mingozzi. Extending ETSI MEC towards stateful application relocation based on container migration. In *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 367–376. IEEE, 2022.

[278] Jakob Schrettenbrunner. *Migrating Pods in Kubernetes*. PhD thesis, Hochschule Darmstadt - Fachbereich Informatik, December 2020.

[279] Minh-Ngoc Tran, Xuan Tuong Vu, and Younghan Kim. Proactive stateful fault-tolerant system for kubernetes containerized services. *IEEE Access*, 10:102181–102194, 2022.

[280] CRIU. A project to implement checkpoint/restore functionality for Linux. `https://github.com/checkpoint-restore/criu`. (Accessed: 2023-11-18).

[281] Schrettenbrunner, Jakob. containerd-cri. `https://github.com/schrej/containerd-cri`. (Accessed: 2023-11-18).

[282] China Academy of Information, Communications Technology (CAICT), et al. Edge native technical architecture white paper. `https://www-file.huawei.com/-/media/corporate/pdf/news/edge%20native%20technical%20architecture%20white%20paper.pdf?la=en`, 2021.

[283] Ignacio D. Martínez-Casanueva, Luis Bellido, Carlos M. Lentisco, and David Fernández. An initial approach to a multi-access edge computing reference architecture implementation using kubernetes. In Honghao Gao, Ramón J. Durán Barroso, Pang Shanchen, and Rui Li, editors, *Broadband Communications, Networks, and Systems*, pages 185–193, Cham, 2021. Springer International Publishing.

[284] Dario Sabella, Alice Li, Hyeonsoo Lee, Luca Cominardi, Qian Huang, Emmanouil Pateromichelakis, Vivek Kashyap, Cristina Costa, Fabrizio Granelli, Walter Featherstone, Faraz Naim, Xu Yang, Hanyu Ding, Sundar Nadathur, Lijuan Chen, Dan Druta, Qi Tang, Bob Gazda, and Gao Chen. Mec support towards edge native design. White Paper No. 55, ETSI, June 2023.

[285] Pedro Escaleira, Miguel Mota, Diogo Gomes, João P. Barraca, and Rui L. Aguiar. Multi-access edge computing as a service. In *Proceedings of the 18th International Conference on Network and Service Management*, CNSM '22, Laxenburg, AUT, 2023. International Federation for Information Processing.

[286] Dario Sabella, Andrew Alleman, Ellen Liao, Miltiadis Filippou, Zongrui Ding, Leonardo Gomes Baltar, Srikathyayani Srikanteswara, Krishna Bhuyan, Ozgur Oyman, Gershon Schatzberg, Neal Oliver, Ned Smith, Sharad D. Mishra, Purvi Thakkar, and Samar Shailendra. Edge computing: from standard to actual infrastructure deployment and software development. White paper, Intel, 2019. (revised August 2021).

[287] Konstantinos Poularakis, Jaime Llorca, Antonia M. Tulino, Ian Taylor, and Leandros Tassiulas. Service placement and request routing in mec networks with storage, computation, and communication constraints. *IEEE/ACM Transactions on Networking*, 28(3):1047–1060, 2020.

[288] Jin Fang, Gongming Zhao, Hongli Xu, Huaqing Tu, and Haibo Wang. Reveal: Robustness-aware VNF placement and request scheduling in edge clouds. *Computer Networks*, page 109882, 2023.

[289] Zhengzhe Xiang, Yuhang Zheng, Zengwei Zheng, Shuiguang Deng, Minyi Guo, and Schahram Dustdar. Cost-effective traffic scheduling and resource allocation for edge service provisioning. *IEEE/ACM Transactions on Networking*, 2023.

[290] Zhengwei Lei, Hongli Xu, Liusheng Huang, and Zeyu Meng. Joint service placement and request scheduling for multi-sp mobile edge computing network. In *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 27–34, 2020.

[291] Weibo Chu, Xinming Jia, Zhiwen Yu, John CS Lui, and Yi Lin. Joint service caching, resource allocation and task offloading for mec-based networks: A multi-layer optimization approach. *IEEE Transactions on Mobile Computing*, 2023.

[292] Ali Gohar and Gianfranco Nencioni. Minimizing the cost of 5g network slice broker. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6, 2021.

[293] Annisa Sarah and Gianfranco Nencioni. Resource allocation for cost minimization of a slice broker in a 5g-mec scenario. *Accepted at Computer Communications*, 2023.

[294] Federico Mason, Gianfranco Nencioni, and Andrea Zanella. A multi-agent reinforcement learning architecture for network slicing orchestration. In *2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet)*, pages 1–8, 2021.

[295] Federico Mason, Gianfranco Nencioni, and Andrea Zanella. Using distributed reinforcement learning for resource orchestration in a network slicing scenario. *IEEE/ACM Transactions on Networking*, 31(1):88–102, 2023.

[296] Chubo Liu, Fan Tang, Yikun Hu, Kenli Li, Zhuo Tang, and Keqin Li. Distributed task migration optimization in mec by extending multi-agent deep reinforcement learning approach. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1603–1614, 2021.

[297] ETSI. ETSI TS 123 304 V17.3.0: 5G; Proximity based Services (ProSe) in the 5G System (5GS) (3GPP TS 23.304 version 17.3.0 Release 17), Jul 2022.