Journal of Information Systems Engineering and Management
2025, 10(38s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

End-to-End Mobility Support Framework for SDN Enabled Multiple-Access Edge Computing (MEC)

Sri Ramachandra L[1], Hareesh K [2], Venkatesh [3]

[1,2] Department of Computer Science and Engineering, Government Engineering College, K R Pet, India
Visvesvaraya Technological University, Belagavi, India
[3] Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bengaluru. India
[1]dsram8388@gmail.com, [2]hareeshk.gec@gmail.com, [3]venkateshr@uvce.ac.in

ARTICLE INFO

Received: 28 Dec 2024

Revised: 14 Feb 2025

Accepted: 21 Feb 2025

ABSTRACT

Multiple-access Edge Computing (MEC) plays a vital role in fifth-generation (5G) cellular networks. MEC ensures reliable and low-latency communication by bringing computational resources and data closer to mobile edge device end- users. It supports the deployment of multi-access edge (ME) applications as executable software running on a virtualization platform, meeting the growing demand for new mobile applications and services. However, end-to-end (e2e) mobility support in MEC remains a research challenge, as seamless service migration (i.e., ME applications) between edge devices must be executed to ensure uninterrupted service. This research proposes a framework that integrates Software- Defined Networking (SDN) and cloud-centric virtualization (e.g., containerizer) with the MEC framework to simplify the coordination and management of multiple mobile-edge-hosts (MEH). The proposed framework supports end-to-end mobility (e2e) to ensure service continuity without disruption when an ME application or mobile user relocates from a source MEH to a target MEH. The SDN-enabled controller proposed in this research coordinates and manages the underlying radio networks to enhance service continuity and quality of service (QoS) metrics, such as latency and bandwidth. The framework was verified and validated using simulation tools. The results demonstrate that the proposed framework effectively supports the end-to-end mobility of ME applications from the source MEH to the target MEH, achieving lower latency and allocating the required bandwidth.

Keywords: Multiple-access Edge Computing(MEC), mobile-edge-hosts(MEH), Software Defined Network, Containerizer, Latency.

1. INTRODUCTION

Today, smart applications are everywhere, and they're demanding a lot from user-portable devices. These new, intelligent applications require high computing power and storage levels, more than most devices can handle on their own [1]. Many applications are designed to offload heavy processing to the cloud to address this issue. While cloud computing offers almost unlimited resources, it can introduce delays because of the distance between the user and the data processing location. For real-time or near-real-time applications, this delay is problematic and makes cloud computing less efficient [2].

Edge computing aims to solve this problem by bringing the cloud closer to the user [3], as shown in Fig. 1. By placing computing resources near mobile base stations, the delay caused by offloading processing is reduced to a level that depends on the performance of the wireless technology being used. Edge computing adds distributed micro data centers to the mobile provider's infrastructure as part of the solution. Mobile communication supports smart applications with increased requirements for bandwidth and low delay and makes edge computing more efficient.

Multi-access Edge Computing (MEC) has been designed to replace or complement cloud-enabled computing and assist the requirements of the telecommunications industry, particularly with the explosive growth of Internet of Things (IoT) devices and the service requirements of new industries. The goal of MEC is to support the most reliable and lower latency communication, minimize traffic to cloud systems, and simplify cloud-enabled computing processes for mobile and IoT devices. This is accomplished by moving computing resources, such as storage and processing capability closer to the data access network in which IoT and mobile devices are located and where data is created [4].

In addition to lowering communication delays, MEC also supports Location-enabled Services (LS) and Radio Network- based Information Services (RNIS), these services make smart network-related decisions for efficient service control and management.

To utilize services offered by MEC in different industries, several significant issues still need to be addressed. For instance, the architecture of MEC for the automotive sector requires end-to-end (E2E) mobility assistance to ensure that services remain continuous when users move from one Host of Mobile Edge (MEH) to another Host of Mobile Edge. To enable this mobility assistance, Mobile Edge (ME) systems shall allow user, application, and service information to transfer without obstacles from one MEH to another [5].

The strategies for service relocation should satisfy the requirements of the application and type of application, whether it's specific to one user or common among many. For specific applications that serve individual users, the application instance needs to move between Mobile Edge Hosts (MEH) as the user travels to ensure continuous service. For shared applications, which serve multiple users (like multicast applications), service continuity can often be managed by moving only the user's context between MEHs, without relocating the entire application.

The main goal of service relocation policies is to reduce downtime. The European Telecommunications Standards Institute (ETSI) suggests "service pre-relocation" as a solution, using vehicle routes to predict and move applications to the next MEH before the user actually transitions to that area [6]. However, service migration involves challenges like predicting the best time to start the relocation, finding a target MEH with enough resources, migrating the application smoothly, updating traffic rules, and creating new communication paths in real-time [5].

To assist and improve service resetting or migration and application relocation in edge computing, this research proposes a Software-enabled Defined Networking (SDN) design that combines cloud-based technologies with executable software (i.e. Docker containers) for virtualization. This setup allows services to be moved smoothly across different edge locations. In the proposed architecture, SDN uses the RNSI and LS (i.e., Radio Network for Information Service (RNIS) and Location Service (LS)) from the Multiple-access Edge (ME) system to anticipate the best time for migration. Further, it chooses the most suitable target Host of MEC (MEH) based on available resources. With help of RNSIS, LS and MEH it performs the necessary handover to the target MEH's network and allocates the required bandwidth application at the new destination or location [7].

The MEC manager plays a vital role in allocating necessary bandwidth to applications. It prioritizes them and helps maintain service quality and resource efficiency across the network.

The proposed architecture, SDN operates as a Mobile Edge coordinator, that facilitates coordination among a hosts of (MEH) and enables efficient decision-making for network routing and management. Incorporating the executable soft- ware (i.e. Docker containers) platform and its associated services into the MEC design introduces valuable features such as stability and application portability, scalability, continuous integration and delivery, and rapid deployment [8][9].

The proposed architecture, SDN operates as a Mobile Edge coordinator, that facilitates coordination among a hosts of (MEH) and enables efficient decision-making for network routing and management. Incorporating the executable soft- ware (i.e. Docker containers) platform and its associated services into the MEC design introduces valuable features such as stability and application portability, scalability, continuous integration and delivery, and rapid deployment [8][9].

The proposed architecture leverages executable software's (i.e., Dockers container) features and introduces a new service, called **Docker Container Registry Service** (DCRS), specifically for service migration within the MEC framework. The DCRS maintains the file system of MEC applications, known as Docker images, which can be instantly executed as MEC applications within executable software's (i.e., Dockers container). With Docker's OS-independent virtualization and portability, these MEC applications can seamlessly move be- tween hosts of MEC(MEH) in real time, enhancing service continuity.

Motivation: Mobile edge computing (MEC) is designed to support applications that need very low latency by offering computing resources close to mobile users. However, implementing MEC in real-world scenarios, especially in managing user mobility, faces challenges. This requires balancing service speed with the costs of moving services between locations. Additionally, managing mobility involves complex interactions across different places and times, which depend on cooperation among users and a clear understanding of user movements and network conditions in advance [10].

Mobile Edge Computing (MEC), an important part of 5G networks, helps improve mobile resources by running heavy applications locally, processing large amounts of data before sending it to the cloud, and providing cloud-like services directly within the radio access network (RAN). By being close to mobile users and using RAN information, MEC also supports context-aware services and enables a wide range of applications [11].

Existing approaches for service migration between MEC nodes typically rely on traditional methods like copying and transferring data but overlook the differences in the underlying networks of MEC hosts (MEH). For low-latency services, the MEH must be located near the Radio Access Network (RAN), and maintaining smooth service continuity requires compatibility between edge clouds managed by different Mo- bile Network Operators (MNOs). Current research also lacks solutions for key aspects of seamless service migration, such as accurately predicting when to start the migration process and identifying target MEH with available computing resources to ensure uninterrupted service.

This research proposes an SDN-based improved MEC architecture that facilitates inter-operator collaboration by sharing the network status of hosts of MEC (MEH) and their underlying networks, such as computational load and channel conditions. This information helps the MEC system select the most suitable MEH for service relocation, thereby improving the user's quality of experience.

Contributions: The contribution of this research work is summarized as follows:

- This research introduces an SDN-based edge computing framework designed to align with the ETSI MEC standard framework, ensuring end-to-end mobility and Quality of Service (QoS) management.
- The proposed framework focuses on enabling seamless service migration between MEC hosts (MEH) across different networks while guaranteeing service continuity.
- This research work proposes a collaboration between Software Defined Network (SDN) and MEC; it illustrates that the SDN paradigm effectively addresses mobility challenges within the MEC environment.
- This research incorporated advanced virtualization using the JFrog Container platform within the MEC architecture. A new MEC service, the JFrog Registry Ser- vice (JRS), was introduced to store and facilitate the relocation of MEC applications
- This research is implemented using simulators SUMO, OMNeT++ and CloudSim, providing the platform to create mobile simulation environments.

Organization of paper: The introduction to the research topic is presented in Section I. A review of relevant research works is provided in Section II. Section III discusses the proposed methodology. Section IV details the proposed resource allocation algorithm for MEH application relocation. The enhanced MEC architecture with mobility support is explained in Section V. Section VI presents the experimental setup, results, and discussion for the V2X simulation scenario. Finally, the conclusion and future work are presented in Section VII.

2. RELATED WORKS

This section provides insight into Multi-Access Edge Computing (MEC), SDD-based MEC for end-to-end mobility and techniques for achieving required QoS. MEC brings the power of cloud computing closer to where data is created and used. Placing computational resources at the network's edge reduces delays, improves service quality, and allows real- time data processing. This approach is useful for applications like IoT, augmented reality, and self-driving vehicles, where quick response times are essential. Therefore, in recent years, MEC has drawn the attention of research scholars, and many researchers have researched the MEC to exploit the various use cases and address challenges.

Authors in [12] proposed a technique that provides End- to-End mobility management in MEC-enabled 5G Networks using SDN and NFV". The authors illustrated how SDN and virtualization in the network (NFV) enable efficient mobility management in MEC for 5G, and their research work focused on minimizing latency and ensuring service continuity for mobile users. The researchers in [13] address service migration in edge computing environments enabled by SDN to maintain low latency. They propose methods for real-time service migration as users move across edge nodes, which is critical for mobile applications. The adaptive resource management and traffic engineering scheme is proposed in [14] to improve the end-user experience despite many frequent mobility events. The research work in [15] addresses service mobility management using SDN in vehicular networks, demonstrating how SDN shall facilitate dynamic resource allocation and handover between edge devices, dynamic resource allocation and handover are vital for ensuring consistent service for mobile users in vehicular environments.

The Lay-Back architecture [16] emphasizes role of SDN in managing MEC resources for improved radio resource sharing and efficient service provisioning, it focus on scalability and flexible resource management for mobile edge applications [16]. The authors in [17] reviewed various MEC handover strategies and management challenges in mobility contexts. The author provide insights into

SDN and shown way the SDN address these issues by offering centralized control and adaptability.

The importance and role of SDN in managing service mobility within MEC architecture for vehicular networks is analyzed in [18]. Authors in [18] proposed scheme that enable seamless transitions and resource allocation to maintain connectivity as vehicles move across network zones, which is critical for 5G and future mobile networks. The researchers in [19] addresses the use of Virtual Network's Functions (VNFs) and resource allocation strategies in SDN-enabled MEC environments, it emphasize on efficient network resource usage and seamless service continuity for mobile edge applications. Authors in [20] proposed SDN based architecture for managing mobility of IoT devices in IoT Networks, the proposed architecture pro- vides a lightweight, non-decentralized approach for managing mobility of IoT devices within MEC environments. It im- proves handoff process, reduces responsiveness and ensures continuous packet delivery in network with multiple mobile nodes. The researchers in [21] explored usage of SDN in industrial wireless sensor networks (IWSNs), researchers in [21] demonstrated how SDN-based frameworks can enhance mobility management in constrained environments by centralizing control and supporting real-time adjustments.

Authors in [22] designed a new handover scheme to manage communication when vehicles move between the coverage areas of two roadside units (RSUs). The proposed scheme ensures a smooth and fast handover on multilane roads with traffic in both directions. The proposed handover scheme uses multiple-access edge computing (MEC) units connected to the RSUs. These MEC units help speed up the handover process by reducing delays and improving efficiency as vehicles move.

By integrating SDN, NFV, SFC and NS technologies to create a unified framework that significantly improves Quality of Service (QoS) across diverse applications [23]. SDN provides centralized control, NFV virtualizes network functions, SFC ensures efficient routing of services, and NS allows dedicated resources for specific use cases. A graph neural network (GNN)-based collaborative deep reinforcement learning (GC- DRL) model to optimize resource provisioning and mitiga- tion strategies for MEC servers under edge DDoS attacks. The model evaluates the trustworthiness of vehicles, designs strategies to mitigate edge DDoS attacks, and ensures reliable resource allocation [24]. The framework is called NFPenabled MEC [25], which uses network function parallelism (NFP) to deliver low-latency services. By running multiple network functions simultaneously, the framework reduces delays and improves the performance of applications running on MEC servers, making it ideal for real-time services.

3. METHODOLOGY

A. Mobile-Edge Computing(MEC) Architecture

ETSI –ISG(industry Specialization group) developed a reference framework [26] to maintain the requirements specified for Mobile Edge Computing (MEC). A reference framework enables seamless interoperability and interaction of ME applications of different computing domains.

The MEC system reference framework outlines the functional components, which include Mobile Edge Hosts (MEH) and the Mobile Edge (ME) management entity, necessary for maintaining and executing MEC applications on the underlying network operator's infrastructure or a subset of it.

Hosts of Mobile Edge (MEH): MEH plays a pivotal role in the reference framework. It comprises a Mobile Edge Platform (MEP) and a network virtualization framework or infrastructure.

- Mobile Edge Platform (MEP) provides storage, computing, and network resources required for running MEC applications.

Journal of Information Systems Engineering and Management
2025, 10(38s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

- The **virtualization framework** includes a data plane responsible for enforcing traffic rules (defined and received from the MEP). The data plane forwards traffic to ap- plications, supports service discovery, and interacts with the proxy/DNS server, 3GPP network, external networks, and local networks.

Mobile Edge Platform (MEP): The MEP is essential for executing MEC applications on a virtualization framework, enabling MEC applications to use and provide services. It also offers its own services. The MEP is responsible for the following functions:

1) Environment for MEC Services: Provides an environment where MEC applications can search, broadcast, consume, and provide MEC services. Records services provided by MEC applications in the MEP and announces available MEC services in other frameworks.
2) Radio Network(RAN) Information Services (RNIS): Provides details about the radio network, including computational load, resource availability, channel status, user location, and corresponding base stations.
3) Traffic and DNS Management: Receives traffic rules from MEC applications, services, or the MEP man- ager and directs the data plane accordingly. Configures DNS/proxy servers based on instructions from the MEP manager. Hosts MEC services and provides access to persistent storage and time information. Handles MEC application instantiation and termination upon requests from the MEP manager.

Mobile Edge (ME) Applications: ME applications are deployed on the virtualization framework of the MEH upon request or configuration, validated and verified by ME man- agement.

- ME applications run on the virtualization framework as virtual machines (VMs).
- They interact with the MEP to use and provide MEC services, as well as to perform lifecycle-related operations (e.g., availability and user state relocation).
- ME applications adhere to rules and requirements, including resource needs, maximum permitted latency, and essential services. These are validated by system-level ME management before deployment.

MEC Administrator (MEA): The MEC Administrator (MEA) is the system-level management component within the MEC framework. The MEA is responsible for:

- System-Level Management: Maintains an overview of the MEC system, including MEH topologies, available resources, and MEC services. Oversees ME application relocation among MEHs based on constraints like latency, resource availability, and service continuity.
- Application Lifecycle Management: Initiates and terminates MEC applications as required. Select the most appropriate MEH for executing MEC applications.
- Package Management and Validation: Maintains a list of packages required for MEC application execution. Validates packages and ensures they meet application requirements and rules. Delegates management of MEC applications to the virtualisation framework manager.

Relocation of Application instance: MEC mobility support enables the relocation of applications among MEH and helps maintain service continuity. Because UE moves from one location to another location in the mobile network, Target MEH is identified as it is more appropriate and useful to execute the ME application and serve the UE. The relocation of the application is initiated or triggered by MEA. An instance of the application running on the source host of mobile edge (MEH) is copied over to target MEH. The operation status of the stateful ME application is transferred between MEHs to continue the service.

Prediction of Relocation and Relocation Management: UE mobility is inevitable as UE moves around the mobile network. To accomplish mobility, the MEH serving must be changed. It is necessary to predict/foreseen the relocation or handover so that the ME application responds to such relocation or handover in an application-specific manner [27]. The accurate prediction of the handover is required to reduce the relocation failure, and relocation failure is directly propositional to the quality of service (QoS) and disruption of the service. The handover failure can be categorized as too late, too early relocation, or reassigned to the wrong MEH. Therefore, accurate and efficient relocation or handover foreseen to the target MEH is a challenging issue in MEC.

MEH Deployment Option Continuous radio access necessitates proper radio station deployment planning and optimization. Several deployment options are available. One such option is co-locating the Mobile Edge Host (MEH) with the LTE radio base node/station (*eNodeB*), while another option is placing the MEH near the LTE radio base station. Deploying the MEH with the LTE radio base node assists in placing services and applications with the *eNodeB*. This deployment option also helps traffic destined for edge devices traverse the Radio Access Network (RAN), thereby reducing latency. How- ever, placing the MEH with the *eNodeB* introduces the complexity of relocating application instances whenever handover is required. On the other hand, placing the MEH in proximity to the eNodeB can serve the entire geographical area, reducing the complexity of frequent application handovers.

This research work considers the deployment option of placing the MEH with the *eNodeB* to enable low latency and high reliability (URLLC) for applications.

4. RESOURCE ALLOCATION ALGORITHM FOR MEH APPLICATION RELOCATION

The researcher in [28] proposed strategies to support service relocation and resource allocation to satisfy low latency and better reliability requirements. The researcher used reinforcement learning method for service -migration and resource allocation real-time scenario, it is efficient service migration and support high mobility. The authors in [29] reviewed research issues of mobility and application relocation in MEC networks.

The mobility support in MEC necessitates an algorithm that allocates resources during service/application relocation with minimal latency and service disruption. The proposed algorithm 1 (i.e., *Resource Allocation Algorithm for MEH Service Migration*) select migration paths and allocate resources to minimize delays and maintain service continuity. The proposed algorithm supports service relocation and resource allocation requirements and satisfies low latency and better reliability requirements.

Algorithm 1 Resource Allocation Algorithm for MEH Application Migration

---

```
Require:
    $S$: Service to be migrated
    $MEH_{Current}$: Current MEH hosting the service
    $MEH_{Target}$: Target MEH for migration
    $R$: Available network resources
    $Q$: QoS requirements (e.g., latency, bandwidth)
    $T_{deadline}$: Maximum allowed migration time  $B_S$: Bandwidth required for service $S$
Ensure: Optimal path and resource allocation for migration
```

```
procedure PathResourceAllocation
    Initialize network topology $G(V, E)$ where $V$ are nodes and $E$ are edges
    Identify all candidate paths $P = \{p_1, p_2, \ldots, p_n\}$
    between $MEH_{Current}$ and $MEH_{Target}$
    Rank paths $P$ based on QoS criteria:
        - Latency
        - Bandwidth availability
        - Hop count
    for all $p \in P$ do
        if Resources on $p$ satisfy $B_S$ and $Q$ then
            Reserve bandwidth $B_S$ along $p$
            Estimate migration time $T_{mig}$ based on $p$
            if $Tmig \leq Tdeadline$ then
                Migrate service $S$ along $p$
                Release resources from $MEH_{Current}$
                return "Migration Successful"
            end if
        end if
    end for
    return "Migration Failed: No Suitable Path Found"
end procedure
```

5. ENHANCED MEC ARCHITECTURE WITH MOBILITY

This section discusses the proposed SDN-based MEC framework for application relocation/handover between hosts of mobile edge (MEH). The proposed framework integrates MEC, SDN, virtualization and cross-platform support to sup- port MEC mobility and relocation of ME applications among MEH affiliated with various networks.

The SDN-based MEC framework is designed based on a reference architecture/model developed by ETSI [26]. Based on the reference model, this research work customized reference model components and functionalities. The customized is an SDN-based MEC framework shown in figure Fig. 1 and discussed in this section.

A. SDN-based mobility-support application in SDN-based MEC framework

The mobility support in MEC necessitates an algorithm that allocates resources during service/application relocation among MEH with minimal latency and service disruption. Dynamic application/service migration is needed to support the mobility feature of MEC. Therefore, this research work proposed the SDN-based mobility support application, which can be placed and executed along with ME applications. The SDN-based mobility support application ensures low-latency relocation operation and uses Location-aware Services (LS) and Radio Network-enable Information Services (RNIS) offered by MEC to gather the updated location of the user or user equipment(UE), existing radio network base station, status/condition of channel, computational load on current MEH and target MEH. The rigorously monitors and records the user or UE position, movement of the user or UE and network condition at MEH. Based on all gathered information and ME application requirements and QoS constraints, SDN-based controller instructs MEA to identify and select an efficient MEH for application/service relocation. Application/service relocation is accomplished through handover performed at network level.

B. SDN-based Relocation Application in SDN-based MEC framework

SDN-based Relocation application for service/application migration is placed on MEH. The SDN-based Relocation application supports low-latency application relocation procedures. SDN-based controller carefully monitors the user or UE movements and underlying radio network through Location- aware Services (LS) and Radio Network-enable Information Services (RNIS) offered by MEC. SDN-based controller (i.e., SDN-based Relocation application) also foresees the necessary relocation of ME applications and initiates the ME application relocation procedure. SDN-based controller (i.e., SDN-based Relocation application) informs the target MEH to initiate an instance of ME application.

SDN-based controller maintains the movements of user or UE through RSSI (i.e., Received Signal Strength Indicator) obtained from various radio network-based base stations in the way of receiving location information through Location- aware Services (LS) and Radio Network-enable Information Services (RNIS). If the value of RSSI received from MEH and various underlying networks falls shorter than the threshold value, then it initiates the ME application relocation procedure at the target MEH and informs the target MEH about this. Traditional methods such as *ZeroMQ* or *MQTT* can be used for inter-process communication between SDN-based controllers and target MEH. This research work uses ZeroMQ, a decentralized, inter-process messaging protocol between SDN-based controller (i.e., SDN-based Relocation application) and target MEH.

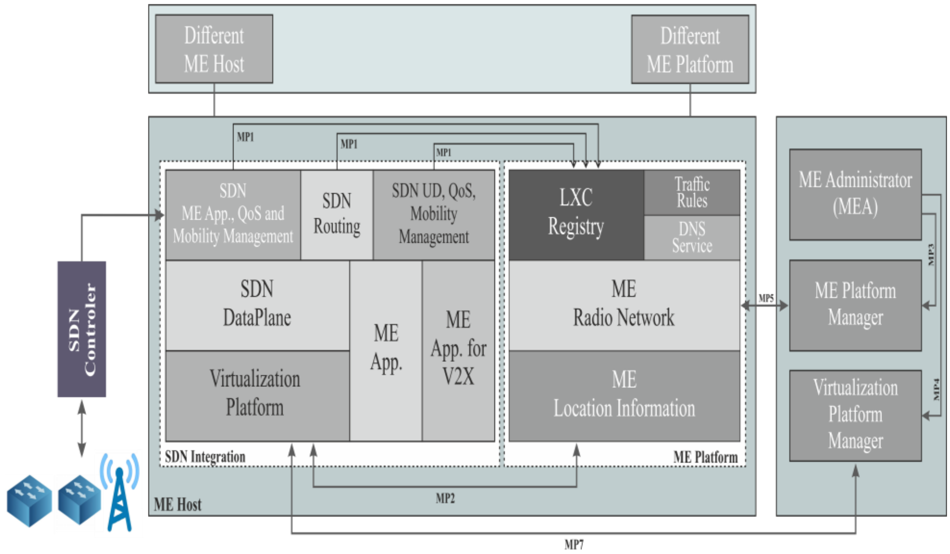


Fig. 1. SDN-based Proposed MEC Framework

C. SDN controller and OpenFlow Switch Bandwidth Manager (SDN-OSBM)

A set of ME applications running in the same MEH will compete for available bandwidth. Every ME application has its own static or dynamic bandwidth requirements. The ME application's static or dynamic bandwidth demand is because it must fulfil the customer's service requirements. An SDN-enabled MEC should support efficient bandwidth allocation, network slicing, and stable throughput for ME applications. The bandwidth allocation becomes further complex when UE or ME applications migrate from one MEH to another MEH. Service continuity of UE or Me application service without disruption necessitates the allocation of required bandwidth at migrated MEH that is associated with another network. This research work proposes a centralised bandwidth allocation approach that adopts an OpenSwitch switch with an SDN controller, a centralised bandwidth allocation approach. OpenFlow switch enables the SDN controller to find a path of packets across the network switches, it consists of flow, group tables and makes use of OpenFlow protocol to interact with the SDN controller.

The OpenFlow Switch with SDN controller is shown in Figure.

D. SND enabled LXC Container

LXC [30] containers provides os-level a virtualization platform. This research uses an LXC container as a substitute for a Virtual Machine (VM). (Because VM increases the complexity and computation overhead of the MEC system). LXC is a lightweight container since it contains ME application-specific or OS-specific packages, it exploits features of the Linux kernel. LXC uses the control group(cgroup) and namespace features of Linux Kernel. The namespace creates a loneliness or isolated loneliness environment for ME application execution, it creates an abstract view of the underlying Linux OS environment (i.e., network resources, OS file system, routing table, packet flow details etc.,). Another feature of Linux kernel is the control group (cgroup) that enforces limits on resources being used and assigns priority to resources like memory, CPU. The LXC is portable, scalable, and easily deployable at MEH. LXC is a Linux-based container that helps in creating and containerization of user applications. The LXC container is executed on MEH and makes use of resources supplied by the virtualization environment for creating and containerizing the user applications. The LXC container is shown in Figure.

E. Container Registry (CR) Service on MEP

This research uses a container registry that contains images. The container images constitute system tools, libraries, and platform settings needed to execute an application. This registry offers the advantages of portability and agility for creating, running, or extending the ME application. In addition to the services mentioned above, it provides a communication protocol (e.g., HTTP API) to aid in delivering the images to the registry engine at the time of interaction/on request. The container registry is running on the MEH mobile-edge platform. Therefore, target or remote hosts can take out required images and interact with the container registry. The remote hosts on target MEH can take out images from source MEH to maintain service continuity. The registry engine uses the pulled images to develop and containerize the application on target MEH. The container registry provides traditional services such as location and RNSI services.

F. LCX Container with Persistent shared Storage for Persistent Application

ME applications are either stateful or stateless. Stateful or persistent applications maintain state or data across interactions or sessions. The container executing the migrated ME application on target MEH has the capability to access the application's data or state, and this feature of the container is being used to maintain the service continuity of the application or UE. Accessing the storage of stateful applications is necessary for the successful operation of ME applications.

LXC containers can store states or application data on storage backends, which provide persistent storage and data that remains intact even if the container is stopped or restarted. The persistent storage in LXC is either done by mapping a directory from the host system to the container to ensure data persistence or by having storage backends create persistent volumes attached to containers or storage stored on the same MEH or multiple-host shared storage.

The directory from the host system to the container will survive until the container expires. If the container exits, then all of the application's stored states or data stored in the directory will be lost. Therefore, this research uses multiple-hosts shared persistent storage, which helps all containers access a shared volume or storage. This multiple-hosts shared volume is necessary because containers must cooperate to provide service continuity and scalability. When multiple MEH run same ME application, this multiple-hosts shared volume helps multiple MEH have access to the application state or data or configuration file. For example, when ME application runs on host A, backend volume is created on host A and stores the application state or data or configuration file. The same backend volume is mounted to another (new) host B on which the ME application is running, and it has access to the application data or state from backend volume. This feature helps in MEC mobility with QoS requirements. The changes made in the backend volume will be mirrored onto all containers on which the same application is running in the MEH. The multiple-host share backend volume for stateful applications is an efficient approach and link of shared backend volume to the container in which the application is running. The container has access to shared backend volume and gets up-to-date application state or data. The link establishment to the container and link disconnection to the container, when it is shifted or moved, are shown in Figure Fig.2.

G. Message Sequences in Iner-MEH to support ME Application Migration.

Figure 3 shows the sequence of messages exchanged be- tween ME-Hosts (MEH) to support the relocation of the ME application and ensure service continuity.

The process begins with the User Equipment (UE) subscribing to the radio network to receive updates related to Radio Network Information and Location Services. This ensures the network has real-time knowledge of the UE's location and radio connectivity details. Such information is critical for enabling efficient handover between different hosts as the UE moves across regions, ensuring seamless service continuity. Once the subscription is active, the system initiates Subscription Updates and Prediction of Handover to Target ME Host(s). This involves updating the UE's subscription information and using predictive algorithms to determine the need for a handover to a Target Mobile Edge (ME) Host. The prediction process evaluates factors such as the UE's current location, movement patterns, and the capability of nearby ME hosts to handle the application workload effectively.

After identifying a capable Target ME Host, the system sends a Request for Instantiation of the Target Application (T-App) Instance at the identified host. The Target ME Host is evaluated based on its available resources, performance metrics, and ability to handle the specific application requirements. Once this is confirmed, the system proceeds to initiate the necessary steps to migrate the application.

Next, the system sends a Request to Pull the ME App (i.e., the containerized application image) from the Source LXC Registry Service to the Target LXC Registry Service. The app image is essential for instantiation at the Target ME Host.

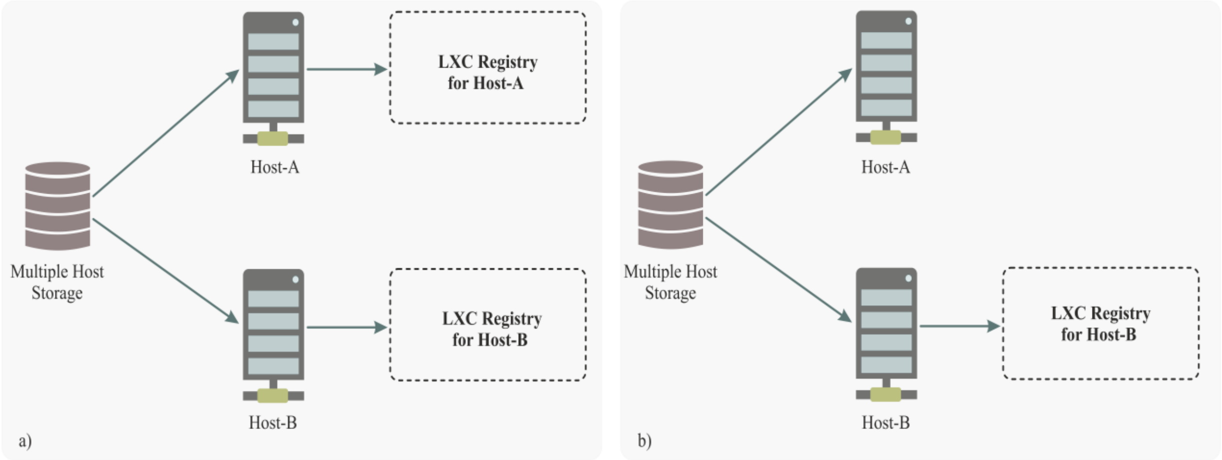


Fig. 2. LCX Container with Persistent shared Storage

Journal of Information Systems Engineering and Management
2025, 10(38s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

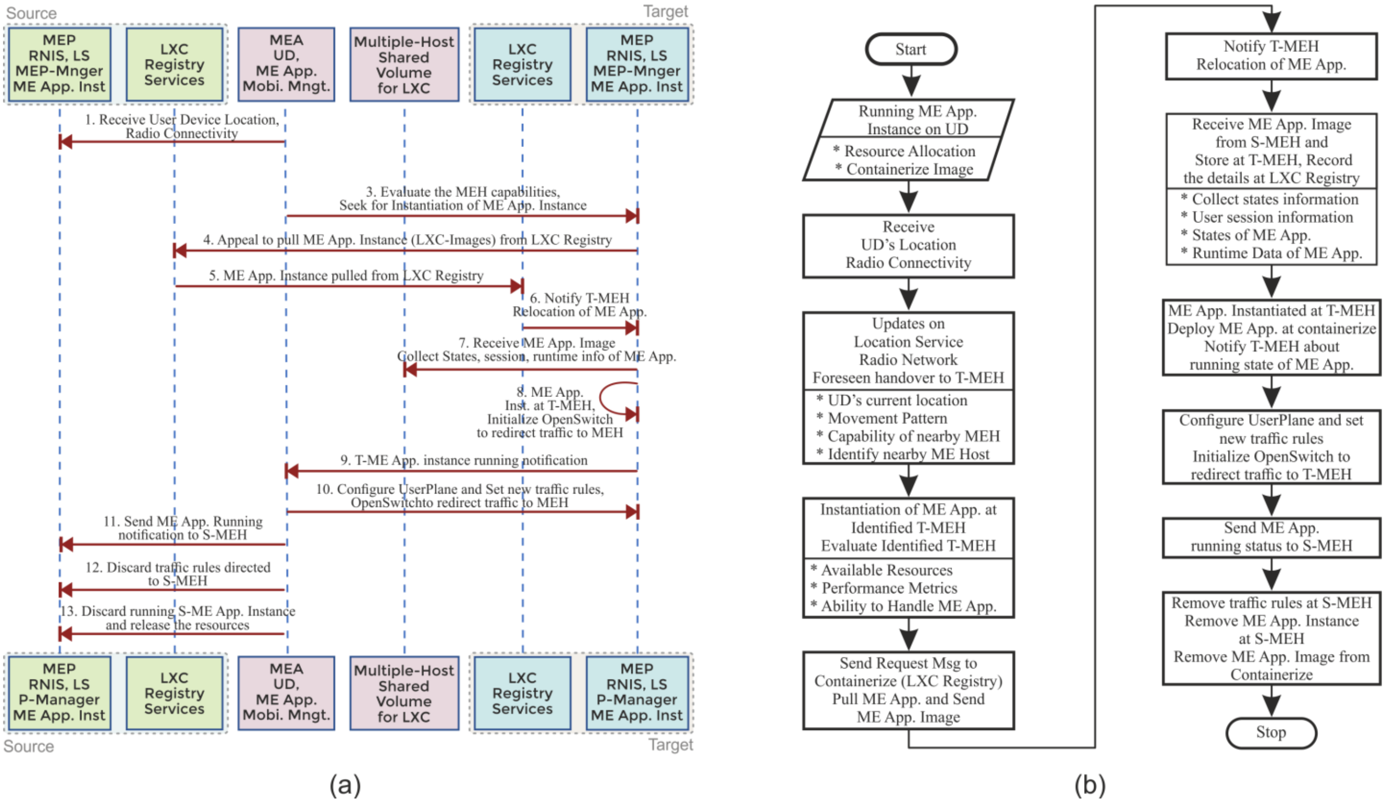


Fig. 3. (a). Sequence of messages exchanged between ME-Hosts (MEH), (b). Flowchart of ME-Hosts (MEH) support for the relocation of the ME application.

This request ensures that the application is transferred securely and efficiently to the target registry service, which acts as a repository for containerized applications.

Once the request is fulfilled, the ME App is Pulled from the Source LXC Registry Service. The image is now available in the Target LXC Registry Service, ready for deployment at the Target ME Host. Following this, a notification is sent to confirm that the ME App has been Relocated to the Target ME Host, ensuring all system components are updated about the app's new location.

The system then performs the Collection of Application State Information from the Source ME Host. This includes gathering essential data such as user session information, application states, and other runtime data to ensure a smooth transition of services. These details are critical to maintaining the continuity of the user experience after the application is moved.

With the necessary data collected, the T-App is Instantiated at the Target ME Host (T-MEH). This step involves deploying the containerized application image at the target location and ensuring it is fully operational to handle user requests. A notification is then sent to confirm that the T-App Instance is Running successfully at the Target ME Host.

Following this, the system proceeds to Configure New Traffic Rules and Instruct the User Plane Function (UPF) or OpenFlow switches to redirect the UE's traffic to the T-App instance. This ensures that user data traffic is seamlessly routed to the application running on the Target ME Host without any service interruptions. A notification is then forwarded to con- firm that the new traffic rules have been applied successfully.

As the transition progresses, the system begins Removing Traffic Rules for Steering UE Traffic Towards the Source ME Host (S-MEH). This step ensures that any previously established traffic routes directing user data to the Source ME Host are deactivated, avoiding unnecessary overhead or routing conflicts.

Finally, the process concludes with the Removal of the Running S-App Instance from the Source ME Host. The application instance, along with its associated container image, is terminated and removed to free up system resources. This cleanup step ensures optimal resource utilization across the mobile edge network.

6. RESULTS AND DISCUSSIONS

A. Simulation Setup

The proposed architecture was simulated on a computer equipped with an Intel Core i9 processor and 64GB of RAM. OpenFlow version 2.5.1 provided the necessary SDN modules for running the SDN controller, while Open vSwitch version 2.5.1 and OpenFlow v1.0 protocols managed data-plane traffic. Core network components, eNodeB, Open vSwitch, and MEH were executed on Linux-based computers.

Table 1 outlines the simulation topology parameters. A two-lane bidirectional road, 5 km in length, was used in the simulations. This road-vehicle scenario included three RSU units, each associated with an MEC server. Detailed specifications of the RSUs and their associated MEC servers are also provided in Table 1.

TABLE I
SIMULATION PARAMETERS

Simulation Parameters	**Values**
Length of Road	3KM
Total Vehicles	2, 3,4,5, 10
RSUs	3
MEC Units	3
Placement of MEC	Equal Distance
Velocity of Vehicle	30Km/h, 40Km/h
Density of Traffic	0.1, 0.3, 0.5 vehicle/m
RSU's transmission range	450m
Transmission level	18 dBm
MEC HDD Storage/	5Gb
MEC CPU speed	2.3 to 2.5 GHz
MEC RAM 2048	2048 Mb
Workload Permitted	50 events/s
Maxim Service rate	12 Mbps
Service Bandwidth Range	50 to 100Mbps
Vehicle Energy	20 to 80 Watt/s

Date Rate	20 Mbps
Channel Max. bandwidth	10Mhz
Sensitivity of Receiver	-90.0 dB
Packet transmission frequency	10MHz
Packet size	200 bytes
OpenFlow Switch version	1.3
Processing delay of SDN Controller	5 µs
Request rate of switch	1600, 3000
Service rate of SDN controller	20,000 request/s
Processing delay of OpenFlow switch	0.5µs
Initial threshold level, SNRRx-th-1	30 dB
Third threshold level, SNRRx-th-3	20 dB

This research focused on evaluating end-to-end (e2e) latency. To achieve this, simulation tools were integrated to analyze e2e delay under various handover scenarios. SUMO [31] was used for simulating urban mobility patterns, OMNeT++ with its 5G extensions [32] managed wireless channels and propagation models, while CloudSim [33] handled resource management in edge computing. LXC/LXD containers were used to emulate hosts, providing secure and cost-effective virtualization infrastructure. These containers are managed using a REST API. The OMNeT++ 5G extensions included libraries for mobile networks (5G and LTE), enabling seamless integration with LXC/LXD containers to evaluate the proposed architecture. Additionally, the research utilized CloudSDN controller features.

The experiments simulated a V2X scenario to assess end-to-end mobility and other Quality of Service (QoS) parameters. Results demonstrated seamless relocation of ME applications or UE services between MEHs, ensuring service continuity. The MEC platform was emulated to enable both V2V and V2X communication. Vehicles were emulated as wireless hosts with interfaces for communication with access points (APs), roadside units (RSUs), and eNodeBs. The radio network was modeled by configuring data rates and transmission ranges for nodes, simulating eNodeB and AP functionalities. These nodes acted as OpenFlow switches, connected to MEHs and a remote cloud.

MEH instances were customized as virtual environments containing LXC containers, interconnected via switches. ME applications were instantiated either in an MEH's LXC container or in a remote cloud. The Ryu SDN controller framework was used, offering APIs to support protocols like OpenFlow. The controller managed traffic flows by accessing and updating flow tables, enforcing data rates, and differentiating services.

To validate the proposed architecture, the LXC containers were deployed on two Linux/Ubuntu-based workstations. Bandwidth varied from 10 Mbps to 100 Mbps to simulate realistic network congestion. The Linux packet scheduler was utilized to mimic real-world MEH deployments and introduce delays between nodes.

B. Scenario for Conduction of Research Experiments

MEC significantly reduces service time by deploying cloud services on edge devices. However, serving mobile edge devices (UE), such as vehicles, requires precise relocation of services or ME applications running on the UE or edge devices. Without meeting latency QoS requirements, the service or application may experience performance degradation. To address this issue, services or applications must be relocated to a new MEH to maintain service continuity and leverage the advantages of MEC. The reference topology for V2X end-to-end mobility support is shown in figure Fig. 4.

In the reference topology, each MEH is a network entity that serves one or more ME applications and operates an SDN controller. The SDN controller uses radio network services, such as RNSI and LS, to gather radio network and location information. This information is crucial for predicting the need to relocate ME applications. The SDN controller identifies a suitable MEH capable of running the ME application and initiates the relocation process in the underlying network of the identified MEH. To facilitate this, the SDN controller employs the Scapy tool to decode packets.

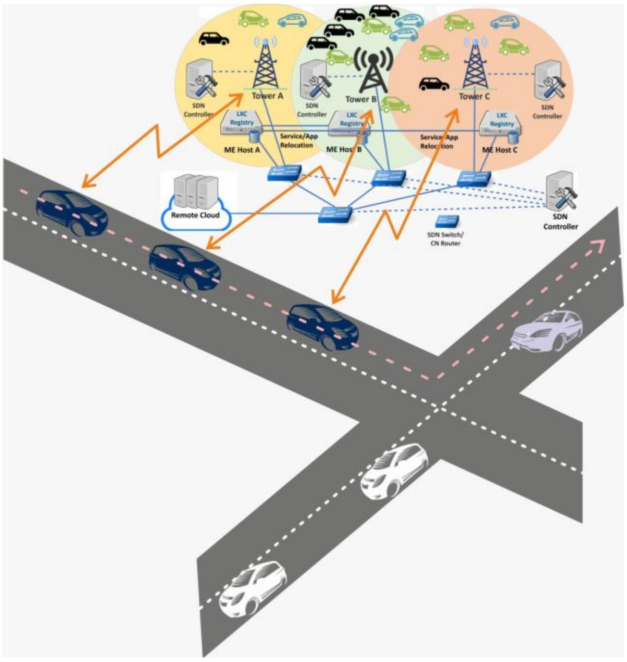


Fig. 4. Reference Topology of V2X scenario

The mobility vehicle or UE periodically sends beacon messages containing RSSI, available network resources, and the computational workload at the serving MEH. If the RSSI value in the beacon message falls below a predefined threshold, the SDN mobility management system triggers the ME application relocation process.

When the SDN controller triggers the migration process, the vehicle, mobile edge device, or UE transitions from a steady state to an inspection state. In the inspection state, the UE calculates the time required to reach a secondary RSSI threshold ($SINR_{Rx-th-2}$), which indicates when the migration process must begin. The bgscan module of Wi-Fi continuously scans RSSI values from APs in the network to support mobility within BSS and ESS. Additionally, the" learn" module in Wi-Fi analyzes channel conditions and operating frequencies over time.

In the proposed architecture, the SDN controller manages flow control by configuring the switch/router's flow and routing tables. It also allocates and manages bandwidth for ME applications to ensure service continuity. As shown in figure Fig. 4, Vehicle A is served by an ME application running on MEH A within Network A. In a heterogeneous radio network, the coverage areas of Network A and Network B overlap.

In the inspection state, the UE or vehicle, equipped with an On-Board Unit (OBU), moves toward a new MEH from the serving MEH A and its underlying radio network. The SDN controllers of both the serving MEH and the target MEH predict the need for application migration based on the observed RSSI of the current radio network. If the current RSSI falls below the threshold and continues to decline, the SDN controller of MEH A sends a relocation request to a target MEH.

In the reference topology shown in Fig. 4, two potential target MEHs (MEH B and MEH C) are considered for service continuity. The SDN controller collects information on computational workloads and available resources from MEH B and
MEH C. Based on this information, the SDN controller selects the most suitable MEH and its associated radio network. From the reference topology, MEH B has a higher computational workload since it serves more vehicles or UEs.

Upon receiving a service migration or relocation request from MEH A, the identified target MEH retrieves the ME application from the LXC container on the source MEH. It also collects the state of the ME application from shared volumes across multiple hosts. Subsequently, the SDN controller initiates the ME application migration process. The entire relocation or handover process begins only after the
SDN controller identifies an appropriate target MEH.

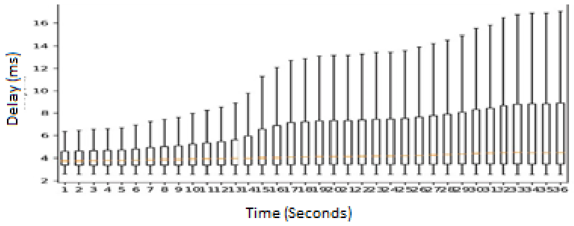
When the UE or vehicle enters the overlapping region of MEH B and MEH C, the SDN controller initiates the ME application relocation or handover to the selected target MEH and its network. The SDN controller reconfigures routing tables and traffic rules to redirect the vehicle's traffic from the source MEH to the target MEH. It also assigns the necessary bandwidth for the ME application using the flow and meter table functionalities of the OpenFlow protocol. To maintain service continuity without disruptions, the ME application running on the vehicle requires a bandwidth of 5 Mbps.

C. Experimental Results Analysis

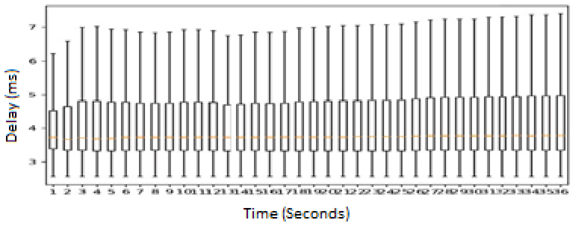
Latency is defined as the time it takes to relocate an ME application from source MEH to target MEH, is critical for service continuity for ME applications. Minimizing latency is essential for ensuring seamless optimal service continuity during ME application migration. Bandwidth is defined as the maximum network data transfer rate, is crucial for efficient application or service migration. High bandwidth reduces transmission time, minimizes latency, and enhances service continuity without disruption.

In our proposed method, when the mobile vehicle or UE moves out of the coverage area of MEH A and its associated network at 30 seconds, it experiences much lower and more stable latency compared to traditional approaches. The reason for achieving lower latency lies in how decisions are made regarding service migration and network handover. In the traditional approach, the service or ME application migration is based solely on signal strength (RSSI, Received Signal Strength

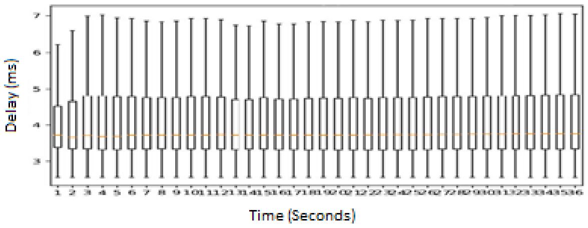
Indicator). However, in the proposed framework, the SDN controller intelligently decides to migrate the ME application or service to MEH C and switches the connection to the underlying network operator (MNO) C. This decision is made based on the computational load and resource availability at MEH C, even though the underlying network at MEH B may have better signal strength (RSSI) than the network at MEH C.



Proposed



Conventional Case-1



Conventional Case-2

Fig. 5. End-to-End Latency Experienced during ME Application Relocation

In the traditional approach, the ME application running on the mobile vehicle or UE remains connected to the remote cloud after the service or ME application is relocated or handed over to the underlying network at MEH B. MEH B is selected based on its higher RSSI values. However, this dependency on remote cloud servers for application relocation or services results in higher latency due to the increased communication distance. In other words, while the RSSI might be strong, reliance on the remote cloud leads to higher latency.

In another traditional approach (i.e., conventional case 2), the SDN controller performs service or ME application migration solely based on RSSI values. This case migrates the service or ME application by considering only RSSI but ignores resource availability. It selects MEH B and its associated underlying network (MNO B) because of better signal strength (RSSI), overlooking MEH B's limited computational capacity.

As the mobile vehicle or UE moves away from the underlying network at MEH B towards another coverage area, the signal strength deteriorates, and latency increases proportionally due to the limited resources available at MEH B. Figure Fig. 6 demonstrates bandwidth allocation at MEH A. MEH B and MEH C when a vehicle moves out of the coverage area of Mobile Edge Host (MEH) A, the SDN controller initiates a mobility management process. Instead of transferring the service to MEH B (which has better signal strength but insufficient resources), it selects MEH C from another operator due to its lower computational load. This ensures the vehicle maintains the required 5 Mbps bandwidth for V2X applications, achieving seamless service continuity.

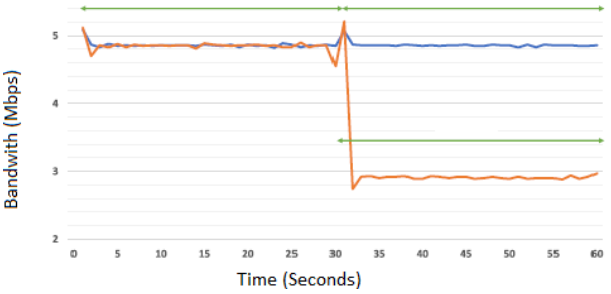


Fig. 6. Bandwidth Allocation by SDN Controller at MEH A, MEH B and MEH C.

In contrast, without inter-operator collaboration, the service is redirected to MEH B. Due to limited resources, MEH B restricts the bandwidth to 3 Mbps using OpenFlow v1.3 metering rules, which fails to meet the application's performance needs.

This approach aligns with research emphasizing SDN's role in optimizing handovers and resource allocation in multioperator environments. By leveraging real-time resource monitoring and traffic control, SDN-based mechanisms facilitate efficient service continuity, particularly in MEC-integrated 5G networks.

Figure Fig. 7 illustrates the changes in bandwidth requests for all ME applications on a congested path. During the time interval from the 1st second to the 39th second, there is minimal demand for bandwidth. It becomes apparent that the bandwidth requirements of the mobility vehicle or mobility device can be satisfied by MEH A. This is clearly depicted by the red and purple curves in Fig. 1, indicating that the bandwidth demand is met by MEH A without compromising service quality or user experience.

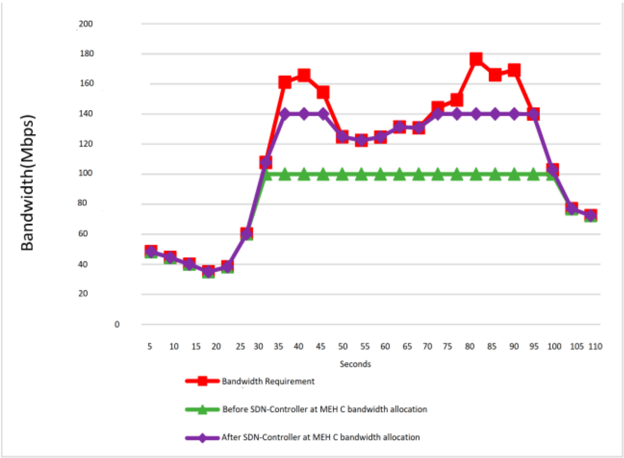


Fig. 7. Bandwidth Allocation by SDN Controller at MEH C.

The purple curve in figure Fig.7 highlights the efficiency of the proposed SDN-based MEC framework. When the ME application is migrated to MEH C and requests bandwidth, the bandwidth request by MEH C is shown as the red curve in Fig. 7. MEH C requires a bandwidth of 180 Mbps at the

40th second (i.e., t = 40). The SDN controller at MEH C responds to this bandwidth request and allocates the necessary bandwidth. The SDN controller at MEH C records this bandwidth allocation as a new traffic rule. This bandwidth allocation is possible at MEH C because MEH C is idle and has the required resources to run the ME application.

When a set of mobility vehicles or UEs arrives (say at t = 105th second) within the coverage range of MEH C and requests bandwidth for their ME applications, the SDN controller at MEH C responds to these requests by allocating the necessary bandwidth and other required resources (e.g., CPU time, memory, etc.). At the same time, the SDN controller reallocates the bandwidth previously assigned to the migrated ME application and records this resource allocation as new traffic rules. The SDN controller at MEH C ensures the relocation of bandwidth to the ME application in such a way that the ME application can continue its service seamlessly. Figure Fig. 8 depicts the service prior-relocation end-to-end latency for relocating container images of 143 MB and 300 MB. Our proposed LXC Registry service achieves lower latency in comparison with traditional approach such as cloud-based migration and copy-and-transfer methods. In the proposed approach, increasing the bandwidth to 100 Mbps further minimize prior-relocation latency.

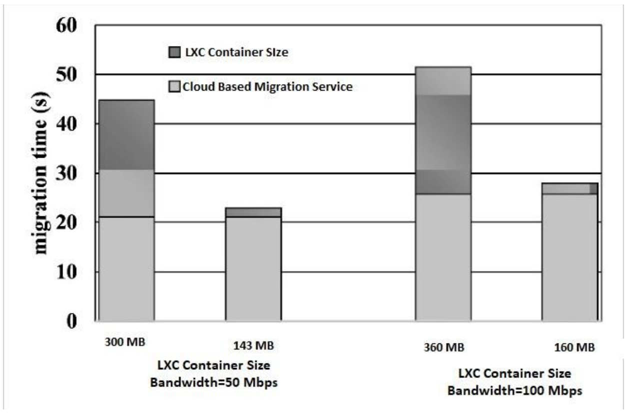


Fig. 8. Experienced Latency during Service Migration.

However, this bandwidth increase does not have a similar impact on traditional methods due to their dependencies on hardware resources. These traditional approaches use time-consuming compression techniques and extraction techniques. Thus, it exports the container image and needs higher bandwidth. The copy-and-transfer techniques export a container image as a TAR file. It uses SCP for Securely transferring the file from one host to another. Further, it import the TAR file to recreate the container image. Another traditional approach cloud-based migration, uses an official cloud service, that manages and distributes containerized images. The proposed approach consistently outperforms these traditional methods in terms of latency.

Moreover, the service prior-relocation latency achieved by the proposed LXC registry is significantly lower than a similar container-based migration approach discussed in [34]. This highlights the efficiency of the proposed solution in reducing service downtime during migration.

7. CONCLUSIONS

Ensuring service continuity without disruption and maintaining a reliable network for mobile edge applications running on automated driving systems or unmanned aerial vehicles is challenging due to

the mobility of edge devices and the interoperability between edge devices or user equipment (UE). This
research proposes an SDN-enabled framework for Multiple-Access Edge Computing (MEC) to support end-to-end mobility with low latency and improve the user experience.

The proposed framework customizes the components and functionalities of MEC by integrating an SDN-enabled controller, a registry service, and cloud-centric virtualization (e.g., containerization). The design of the framework aligns with the MEC reference architecture.

The proposed framework is verified and validated through simulation tools in a V2X (vehicle-to-everything) scenario. The experimental results demonstrate that the framework achieves seamless ME application migration from the source MEH to the target MEH with lower latency while satisfying bandwidth requirements.

As future work, the deployment of the SDN-enabled MEC framework on real-time edge devices will be tested to further support end-to-end mobility.

REFERENCES

[1] S. Abolfazli, Z. Sanaei, A. Gani, F. Xia, L. Yang, Rich Mobile Applications: Genesis, Taxonomy, and Open Issues, *Journal of Network and Computer Applications*, vol. 40, pp 345-362, 2014.

[2] X. Ma, Y. Zhao, L. Zhang, H. Wang, L. Peng, When Mobile Terminals Meet the Cloud: Computation Offloading as the Bridge, *IEEE Networks*. vo. 27, no.5, pp. 28-33,2013

[3] M.M. Satyanarayanan, The Emergence of Edge Computing, *IEEE Computer* vol.50 no.1, pp.30-39, 2017.

[4] A.Yousefpour,C.Fung,T.Nguyen,K.Kadiyala,F.Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete survey, *Journal of Systems Architecture*, vol. 98, pp. 289-330, Sep. 2019.

[5] MEC ETSI ISG for Multi-Access Edge Computing (MEC) and ETSI Industry Specification Group (ISG), Mobile edge computing (MEC); end-to-end mobility aspects, *ETSI*, Sophia Antipolis, France, Tech. Rep. GR MEC018, Oct. 2017.

[6] Raghavendra Nayaka, P., Ranjan, R. An Efficient Framework for Algorithmic Metadata Extraction over Scholarly Documents Using Deep Neural Networks. SN COMPUT. SCI. 4, 341 (2023). https://doi.org/10.1007/s42979-023-01776-3.

[7] MEC ETSI ISG for Multi-Access Edge Computing (MEC) and ETSI Industry Speciation Group (ISG), ”Multi-access edge computing (MEC); phase 2: Use cases and requirements”, *ETSI*, Sophia-Antipolis, France, Tech. Rep. GS MEC 002, Oct. 2018.

[8] Kumble, L., Patil, K.K. An Improved Data Compression Framework for Wireless Sensor Networks Using Stacked Convolutional Autoencoder (S-CAE). SN COMPUT. SCI. 4, 419 (2023). https://doi.org/10.1007/s42979-023-01845-7

[9] He, Q., Zhang, F., Bian, G. et al. Real-time Network Virtualization based on SDN and Docker Container. Cluster Computing, vol. 26, pp. 2069–2083, 2023. https://doi.org/10.1007/s10586-022-03731-y.

[10] F. Guo and M. Peng, ”Efficient Mobility Management in Mobile Edge Computing Networks: Joint Handover and Service Migration,” *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 18237-18247, 15 Oct.15, 2023, doi: 10.1109/JIOT.2023.3279842.

[11] Suresh M et al.,, “Analysis of Score Level Fusion of Biometric Features” International Journal of Electrical and Electronics Research , Volume 10, Issue 2, Pages: 250-255, e-ISSN: 2347-470x, DOI: https://doi.org/10.37391/IJEER.100233, 2022.

[12] E. Schiller, N. Nikaein, E. Kalogeiton, M. Gasparyan, T. Braun, CDSMEC: NFV/SDN-based Application Management for MEC in 5G Systems, *Journal on Computer Networks*, Vool. 135, pp. 96-107, 2018.

[13] Zheng, G., Navaie, K., Ni, Q. et al. Energy-efficient Secure Dynamic Service Migration for Edge-based 3-D Networks, Journal on Telecommunication Systems, vol. 85, pp. 477–490, 2024. https://doi.org/10.1007/s11235-023-01094-2.

[14] Chunlin Li, Kun Jiang, Youlong Luo, Qos-aware Mobile Service Optimization in Multi-Access Mobile Edge Computing Environments, *Journal on Pervasive and Mobile Computing*, Vol. 85, https://doi.org/10.1016/j.pmcj.2022.101644, 101644, 2022.

[15] S. D. A. Shah, M. A. Gregory, S. Li, R. d. R. Fontes and L. Hou, "SDNBased Service Mobility Management in MEC-Enabled 5G and Beyond Vehicular Networks," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13425-13442, 1 Aug.1, 2022, doi: 10.1109/JIOT.2022.3142157.

[16] P. Shantharama, A. S. Thyagaturu, N. Karakoc, L. Ferrari, M. Reisslein and A. Scaglione, "LayBack: SDN Management of Multi-Access Edge Computing (MEC) for Network Access Services and Radio Resource Sharing, *IEEE Access*, vol. 6, pp. 57545-57561, 2018, doi: 10.1109/ACCESS. 2018.2873984.

[17] S. R. Alkaabi, M. A. Gregory and S. Li, "Multi-Access Edge Computing Handover Strategies, Management, and Challenges: A Review," *IEEE Access*, vol. 12, pp. 4660-4673, 2024, doi: 10.1109/ACCESS. 2024.3349587.

[18] S. D. A. Shah, M. A. Gregory, S. Li, R. d. R. Fontes and L. Hou, "SDNBased Service Mobility Management in MEC-Enabled 5G and Beyond Vehicular Networks," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13425-13442, 1 Aug.1, 2022, doi: 10.1109/JIOT.2022.3142157.

[19] N. Kiran, X. Liu, S. Wang and C. Yin," VNF Placement and Resource Allocation in SDN/NFV-Enabled MEC Networks," *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW),* Seoul, Korea (South), 2020, pp. 1-6, doi: 10.1109/WCNCW48565.2020.9124910.

[20] S. D. A. Shah, M. A. Gregory, S. Li, R. d. R. Fontes and L. Hou, "SDNBased Service Mobility Management in MEC-Enabled 5G and Beyond Vehicular Networks," *IEEE Internet of Things Journal,* vol. 9, no. 15, pp. 13425-13442, 1 Aug.1, 2022, doi: 10.1109/JIOT.2022.3142157.

[21] L. L. Bello, A. Lombardo, S. Milardo, G. Patti and M. Reno, "Experimental Assessments and Analysis of an SDN Framework to Integrate Mobility Management in Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics,* vol. 16, no. 8, pp. 5586-5595, Aug. 2020, doi: 10.1109/TII.2020.2963846.

[22] Monir N, Toraya MM, Vladyko A, Muthanna A, Torad MA, El-Samie FEA, Ateya AA. Seamless Handover Scheme for MEC/SDN-Based Vehicular Networks, *Journal of Sensor and Actuator Networks*. 2022; 11(1):9. https://doi.org/10.3390/jsan11010009

[23] Suresh M et al., "Performance Analysis of Multimodal Biometric System Using LBP and PCA", International Conference on Recent Trends in Electronics and Communication, **DOI:** 10.1109/ICRTEC56977.2023.10111925.

[24] Y. Deng et al, Resource Provisioning for Mitigating Edge DDoS Attacks in MEC-Enabled SDVN, *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24264-24280, 1 Dec.1, 2022, doi: 10.1109/JIOT.2022.3189975.

[25] M. Liu, G. Feng, Y. Sun, N. Chen and W. Tan, "A Network Function Parallelism-Enabled MEC Framework for Supporting Low-Latency Services," in *IEEE Transactions on Services Computing,* vol. 16, no. 1, pp. 40-52, 1 Jan.-Feb. 2023, doi: 10.1109/TSC.2021.3130247.

[26] MEC ETSI ISG for Multi-Access Edge Computing (MEC) and ETSI Industry Specification Group (ISG), "Multi-access edge computing (MEC); framework and reference architecture," *ETSI*, Sophia-Antipolis, France, Tech. Rep. GS MEC 003, Jan. 2019.

[27] MEC ETSI ISG for Multi-Access Edge Computing (MEC) and ETSI Industry Specification Group (ISG), “Mobile edge computing (MEC): End to End Mobility Aspects,” *ETSI*, Sophia Antipolis, France, Tech. Rep. GR MEC 018, Oct. 2017.

[28] Chunlin Li, Lei Zhu, Weigang Li, Youlong Luo, ”Joint edge caching and dynamic service migration in SDN based mobile edge computing”,*Journal of Network and Computer Applications*, Volume 177, 2021.

[29] Z. Liang, Y. Liu, T. -M. Lok and K. Huang, ”Multi-Cell Mobile Edge Computing: Joint Service Migration and Resource Allocation,” in *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 5898-5912, Sept. 2021, doi: 10.1109/TWC.2021.3070974.

[30] Ameya, ”Unveiling the Differences: LXC vs Docker – An In Depth Comparison”, Published on 5th August, 2023, Accessed on 10th May 2024, https://www.redswitches.com/blog/lxc-vs-docker/linuxcontainers-lxc-a-closer-look.

[31] P. Lopez Alvarez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, P. Wagner, E. Wiesner, Microscopic traffic simulation using SUMO, *IEEE Intelligent Transportation Systems Conference (ITSC*), 2018.

[32] Banu, W., Kumar, R.S. *et al.* Eye Hypertension Disorder Through Ensemble Frame Networks: A Novel Approach for Early Identification and Segmentation. *SN COMPUT. SCI.* **5**, 1068 (2024). https://doi.org/10.1007/s42979-024-03427-7.

[33] S. Filiposka, A. Mishev, K. Gilly, ”Mobile-aware Dynamic Resource Management for Edge Computing, Transactions on *Emerging Telecommunication Technology*, vol. 30, no.6, 2019.https://doi.org/10.1002/ett.3626.

[34] Monir, Nirmin, et al. ”Seamless handover scheme for MEC/SDN-based vehicular networks.” *Journal of Sensor and Actuator Networks*, 11.1, 9,2022.