2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS) | 978-1-6654-7706-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/COMSNETS56262.2023.10041332

A 5G OpenAirInterface (OAI) Testbed with MEC: Deployment, Application testing and Slicing Support

Priyal Thakkar
Bharti School of Telecom Technology & Mgmt
Indian Institute of Technology Delhi
New Delhi, India
priyalthakkar1997@dbst.iitd.ac.in

Shashvat Sanadhya
Bharti School of Telecom Technology & Mgmt
Indian Institute of Technology Delhi
New Delhi, India
shashvat30@gmail.com

Pimmy Gandotra
Bharti School of Telecom Technology & Mgmt
Indian Institute of Technology Delhi
New Delhi, India
pimmy.gandotra@gmail.com

Brejesh Lall
Department of Electrical Engineering
Indian Institute of Technology Delhi
New Delhi, India
brejesh@ee.iitd.ac.in

***Abstract*—Realizing the potential of multi-access edge computing (MEC) is essential to achieve low latency and high reliability targets. By supporting computation and storage close to the users, value services can be delivered to the subscribers. The MEC architecture proposed by ETSI caters key requirements for the next generation applications. The OAI kind of initiatives are increasingly becoming the vehicle for 5G deployments, allowing network customization and real-time monitoring. Seeking a flexible architecture for wireless networks, this paper puts forth a novel 5G MEC-OAI testbed, adhering to the existing standards. While leveraging the standards, the proposed setup is easy to deploy because of virtualized infrastructure. A video streaming application has been demonstrated on this architecture, addressing object detection scenarios. Additionally, the testbed can be easily adapted to support network slicing.**

***Index Terms*—MEC, OAI, testbed, network slicing, video streaming**

I. Introduction

The 5G wireless communication networks (WCNs) are expected to rule the era of cellular communications, with continuous evolution and amendments in component technologies. These networks also serve as the building blocks for the 6G networks. The diversity of 5G applications is accompanied by intensive compute and storage requirements. Leveraging a platform for reliable services at the edge of the network, MEC technology is providing new opportunities for operators and subscribers.

An advancement of cloud computing, MEC, enables real-time applications in an efficient manner while meeting low latency and high reliability targets. The standards development in Edge Computing is headed by ETSI [1], [2]. The architecture proposed by the ETSI comprises of the MEC Host (MECH), along with the other MEC components required to run applications on the MEC system. 3GPP also addresses the Edge Computing architecture, however, from the application perspective [3].

The MEC being a complex setup, requires appropriate operational and deployment support. Considering the same, this paper proposes a novel MEC-OAI 5G testbed. The OAI-5G Core Network (CN) project group's objective is to deliver a fully functional, 3GPP-compliant 5G Standalone (SA) CN implementation. OAI 5G CN is created and implemented in a flexible manner that makes it simple to adapt to the requirements of various 5G use cases. With the help of expert testers, commercial gNBs (with COTS UE), and open-source RAN simulators, all the characteristics of the OAI 5G CN components are regularly tested. OAI 5G CN now provides fundamental connections, registration (UE registration, deregistration, and service requests), and session management processes (PDU session establishment, modification, and release). Other capabilities supported by OAI 5G CN include N2 Handover, HTTP/2, FQDN support, Paging, and Network Slicing (partially supported). The OAI 5G CN can accommodate multiple UEs and many PDU sessions simultaneously (please refer to this tutorial for more information), along with support for various use-case scenarios.

Tailoring to the needs of the demanded applications, network slicing provisions virtual networks (VNs) over a common infrastructure and can be adapted by the proposed testbed. The integration of network slicing technology into the MEC has been investigated by 3GPP [4] and is evolving further.

A. Related Work

The rising demand of subscribers and their diverse requirements seek a flexible network architecture that has led to the integrated use of open-source 5G technologies. A MEC testbed has been discussed in [5], following the specifications from ETSI. The authors in [6] discuss gaps in MEC systems, wherein the role of the components of MEC is illustrated for supporting network slicing, further elaborated with various applications. An open-source testbed has been proposed in [7], addressing the security threats at the edge. The authors in [8] use a testbed for investigating the benefits of network slicing, as well as ultra-low latency communication, for eHealth services. A virtualized infrastructure for 5G networks is proposed in [9], orchestrated via Mosaic 5G and Kubernetes, deployed on 4G standards.

Most of the existing testbeds are not open source, and lack standards compliance. Though there has been a rise in the open-source testbed proposals, integration with the edge is still under development phase. These lack robustness, opening up opportunities for further investigation.

B. Contribution

In the endeavour towards building up an OAI-MEC testbed for 5G and beyond wireless communication networks, and motivated from the existing work, the key contributions of this paper are:

- An end-to-end testbed is developed by integrating ETSI-compliant MEC and OpenAirInterface (OAI) infrastructure, that triggers integration of multi-standardization support in a single architecture.
- An open-source robust 5G testbed, which is easily deployable, portable and hardware independent
- A demonstration of real-time applications can been put forth over the proposed system. A video streaming application addressing object detection scenarios has been demonstrated to illustrate its operability
- Support for network slicing has also been incorporated into the testbed.

II. Functional Design, Architecture and Standards

Participation from open-source communities can result in new innovations to improve continuing standardisation, alignment, and technological accessibility. This section discusses the potential synergy between distinct streams of standards and their respective deployments. Creating a standardised end-to-end architecture from a set of mutually beneficial components is the best possible conclusion for the industry as well as academia. It is crucial to guarantee alignment, harness synergies, and provide consistent techniques and tools for developers in order to speed up time-to-market and encourage industry adoption. A 5G OpenAirInterface (OAI) Testbed with MEC: Deployment, Application testing and Slicing Support

A. MEC Standards

Managing the network dynamically and provisioning services with compute and virtualization capabilities for MEC has been standardized by ETSI and 3GPP.

1) ETSI MEC Architecture: The ETSI MEC system is composed of different architectural components. The communication flow among the components is enabled via distinct interfaces. The reference architecture is discussed in [10]. Following are some key entities of the system:

a) MEC Orchestrator (MECO): In charge of the topology, deploys hosts (MECH), resources, and MEC services for the entire MEC system.

b) MEC Platform Manager (MEPM): The MECO is instructed of the pertinent application-related events, service authorizations, traffic rules, and DNS configuration by the MEPM, which also manages the life cycle of applications. Its functionality is incorporated into MECO in this work.

c) MEC Platform (MEP): The MEC services or applications can discover, register, and provide MEC instantiation inside the virtual infrastructure provided by the MEP.

d) MEC Host (MECH): This system consists of the MECP and a virtualization infrastructure that gives the MEC applications (MEC apps) computing, storage, and network resources.

e) MEC app: It operates as a virtual machine on the MECH, communicates with the MECP to use and supply MEC services, and supports processes relevant to the application life cycle.

f) Operations Support System (OSS): Receives requests from device applications for instantiation or application termination via the Customer Facing Service portal (CFS). The feature is included in MECO.[1]

g) User life cycle management (UALCM): It gives device applications the ability to ask for the instantiation, termination, and movement of applications within the MEC system. Thus, UALCM manages the entire lifecycle of the UE context.

The MEC Interfaces between the MEC system entities are explained below; three groups of reference points (interfaces) are established:

a) Mp interface: The above interfaces are for MECP-related functionalities. There are three reference points in this context: Mp1, Mp2 and Mp3. The Mp1 interface implements service registration and discovery, including DNS and traffic rule registration and updating. It also established communication between the MEC app and MECP. Mp2 authorises communication between the MECP and the data plane. Mp3 executes communication between the MEC platforms.

b) Mm interface: The group of interfaces provides management-related functionalities. There exist Mm1 to Mm9 interfaces [10], between distinct entities, managing services, applications and resources[2].

c) Mx interface: The above interfaces are related to external entities. Two reference points exist: Mx1 and Mx2 [10], which request the running of applications in the MEC systems and

[1] However, it is not deployed in work being done right now.

[2] This work incorporates the mm9 interface between UALCM and MECO as well as between MECH and MECO. All other management interfaces, including Mm1, Mm2, Mm3, Mm4, Mm5, Mm6, and Mm7, are combined into Mm9.

moving applications in/out of the system, respectively[3].
The information flow in the MEC system is elaborated in [5].

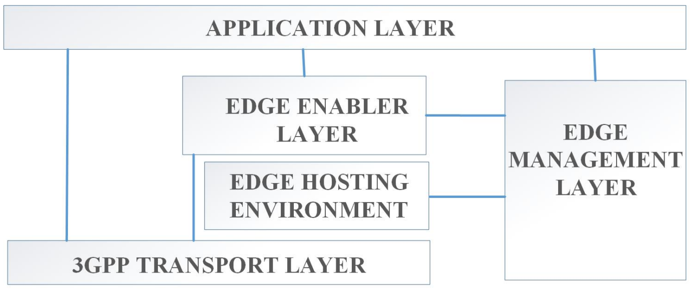

Fig. 1: 3GPP Edge Computing capabilities: An Overview

2) 3GPP Architecture: Based on the application needs, the application clients (ACs) must locate and be able to connect with an appropriate application server in the edge data network (EDN). 3GPP supports the capabilities for edge computing, which are illustrated in Fig. 1. Of all the edge compute capabilities, the edge enabler layer is discussed in [3], transport layer is discussed in 3GPP TS 23.401 [11] and 3GPP TS 23.501 [12], edge management layer is discussed in 3GPP TS 28.538 [13]. The edge hosting environment does not lie within the scope of 3GPP.

The local data network is the EDN, which also contains the Edge Application Server (EAS) and Edge Enabler Server (EES). The EES configurations are provided by the Edge Configuration Server (ECS), which also includes details of EES being hosted by the Edge Data Server (EDS). The Edge Enabler Client (EEC) and ACs are contained in the UE. EES, ECS, and EAS communicate with the 3GPP core network.

Some key differences exist between the ETSI and 3GPP architectures and have been stated in Table I.

3) ETSI and 3GPP Architecture: Both ETSI ISG MEC and 3GPP have developed their own architectures for edge computing within the domains of their respective organizations. Their shared objective is to establish an accessible and standardised IT service environment for hosting and supporting third-party applications in edge environments. MEC Platform/Edge Enabler Server and MEC Applications/Edge Application Servers, respectively, are the core components of the ETSI ISG MEC and 3GPP SA6 architectures. There is a great degree of synergy apparent in the two architectures on these aspects, and in the information carried between these functional entities. The synergized architecture for the Mobile Edge Cloud, supported by ETSI and 3GPP, has been elaborated in [14].

B. OpenAirInterface (OAI) Standard

For the proposed testbed, we require a relevant software implementation of the 5G stack that mimics the actual mobile cellular system and is compliant with the 3GPP standards. It should provide a complete software implementation of the 5G protocol stack, i.e., SDAP, PDCP, RLC, MAC, PHY, and RRC layers for eNodeB (gNB) and User Equipment (UE). For the Core Network (CN), it should split the traditional Evolved Packet Core (EPC) into the user plane and control plane components that should imitate the 5G CN components.

The OAI platform provisions complete implementation of the cellular networks that are 3GPP standard compliant [15]. It is also beneficial for testing real-time scenarios. The OAI is a flexible and reconfigurable platform, supporting OAI RAN, core, and other projects. For the setting up of the OAI testbed, the deployment scenario can be obtained from their official website [16].

[3]The Mx2 interface is implemented in this work, between the UE and the UALCM

III. Experimental Setup

This section describes the experimental setup of the 5G OAI core and MEC (ETSI-compliant) and their integration. The setup is depicted in Fig. 2. The prerequisites for the setup are illustrated in Table II.

A. MEC System Setup

The MEC entities are deployed with the help of VMware machines. The individual virtual machines were created for the components shown below in the VMware virtualization tool:-

- UALCM
- MECH
- MECO

Python Flask is used to implement each MEC entity within the VM machines. The API names are given on the basis of MEC interfaces. The UE context, app lists, MEC platform, and UALCM details are maintained in the MySQL database.

B. 5G OAI Testbed Setup

5G OAI testbed setup is a Docker container-based infrastructure comprising the following entities:

- AMF
- SMF
- UPF
- NRF
- gNB

Each entity is a single Docker container. OAI provides a UE Docker container, which is connected to the UPF via a GTP tunnel. The MEC device app (client application) is merged with the OAI UE Docker container. Both the OAI and MEC configurations are running on a Dell machine with 32 GB of RAM, a 1 TB hard drive, and an i9 processor. The system setup is depicted in Fig. 2.

The interaction between the device app and the MEC System is depicted in Fig. 3. The MEC app is instantiated and on-boarded, as shown in the figure, while the information about the application is fetched, instantiated, and returned to the device app. It is made up of the following queries and responses:

- A GET request to locate and retrieve a list of available apps with the specified parameters, followed by a response.

TABLE I: Comparision between ETSI and 3GPP standards.

Features	ETSI MEC	3GPP Edge Setup
Functional entities	UE, UALCM, OSS, MEC orchestrator, MEC host, MEC platform, Management plane.	Edge application server(EAS), Edge Enabler Server (EES), Application Client (AC), Edge configuration server (ECS), Edge Enabler client (EEC).
Interface (reference points)	Mx (for external entities), Mm (for management tasks), Mp (for communication with MECP).	EDGE-1 (Between EEC and EES), EDGE-2 (between the EES and the 3GPP Core Network functions), EDGE-3 (Between EAS and EES), EDGE-4 (Between EEC and ECS), EDGE-5 (Between AC and EEC), EDGE-6 (Between EES and ECS), EDGE-7 (Between the EAS and the 3GPP Core Network functions), EDGE-8(ECS and the 3GPP Core Network functions), EDGE-9 (Between EES and EES).
Interaction with 5G core(5GC)	MECO interacts with 5GC via N6 interface, MECH connects with UPF of 5GC.	EAS, ECS, and EES connect to the 5GC via respective interfaces.
Orchestration Entity	MECO	No orchestrator as such. However, EES and ECS together maintain UE registration, life cycle application etc.
Life cycle management entity	User application life cycle management(UALCM).	No such entity.
MEC Application instantiation	Inside MECH, the MECP deals with application instantiation.	Application client (AC) connects with EAS to complete application instantiation.

- Send a POST request for app context generation and MEC Host selection for a response.
- PUT request; optional if there is a modification to the callback reference.
- DELETE request to stop an application that is currently running.

C. *Integration with OAI 5G Core*

OAI offers three distinct rent options for the 5G core infrastructure [16]. The setup in Fig. 2 illustrates a single UPF (OAI Minimal setup), which is used to execute the integration between the ETSI MEC system and OAI 5G core. A single UPF solution is a typical deployment solution where all the data is sent to a single edge point. Following are some general integration concerns and actions:

- gNB provides a strong connection between the UE OAI Docker container and the AMF and UPF Docker containers.
- Connection is established between the UE and UPF through a GTP tunnel via gNB.
- To complete the MEC app instantiation process, MECO registers the UE context and provides the destination address (reference URL) to UE.

TABLE II: Pre-requisites for Testbed.

Name of Entity	Software/Virtualization Platform	Host OS
OAI 5G Core setup	Docker Engine, Docker Compose	Ubuntu 18.04.4
MEC	VMware	Ubuntu 18.04.4
Database	MYSQL	Ubuntu 18.04.4

- Based on the destination IP address, UPF will then direct this to the corresponding DNAI (EDGE) (over the N6 interface).

With rising subscriber counts and corresponding demands for different classes of applications with stringent latency, data rate, and reliability requirements, the creation of network slices has been identified as an efficient and precise scheme for ensuring high performance. This testbed setup aims to provide end-to-end slices for supporting the new applications.

The designed ETSI-compliant 5G testbed will result in an easily deployable tool that will allow organisations or academic institutions to concentrate on feature development in particular. The testbed is portable across many platforms, enabling the community to test the features in multi-vendor settings quickly and effectively. Because the entire configuration is virtualized, the resulting testbed will be independent of the hardware environment.

IV. Application Deployment and Testing

With low-latency applications stipulating the current wireless network scenario, MEC will serve as a candidate technology for the same. The developed 5G testbed is appropriate for numerous applications that require high processing power, intensive storage, and data offloading strategies. The developed 5G testbed is shown to work by using the video analytics application, but the testbed can be used to deploy other types of applications as well.

A. *Video Streaming Application Deployment*

Due to the moderate computing capability and low latency features of the MEC server, it can be used to detect a

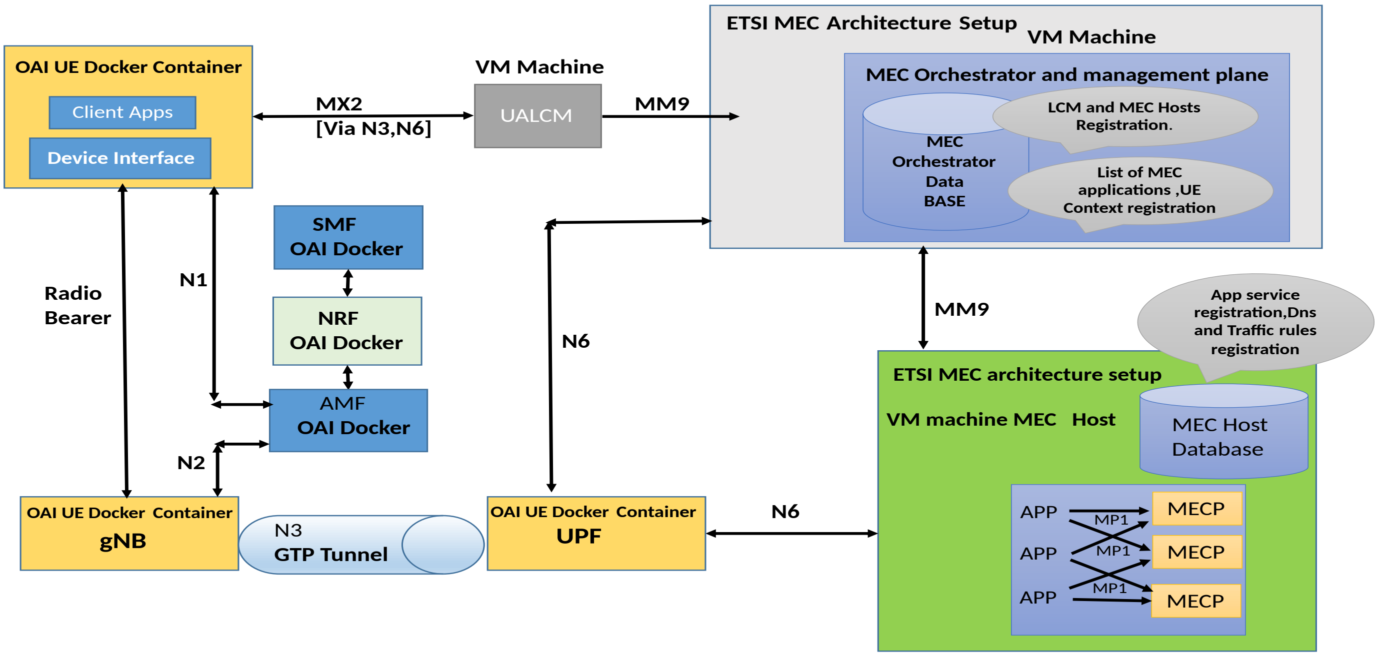

Fig. 2: Architecture setup of 5G MEC-OAI testbed, using Virtual machines and Docker container.

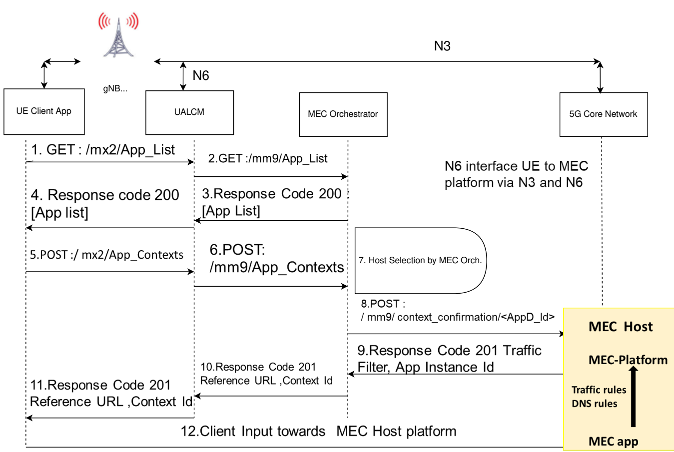

Fig. 3: Information flow for 5G MEC-OAI testbed.

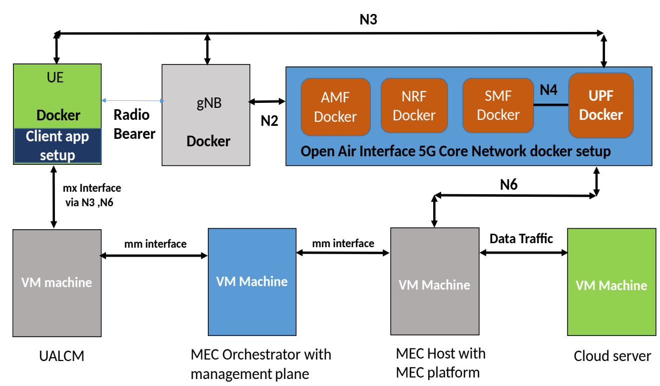

Fig. 4: 5G MEC-OAI testbed deployment setup.

few critical features that are essential in the video analytics applications. The aforementioned capabilities can be used to detect accidents or traffic offences from the videos. Hence, the few components of video analytics applications that require immediate processing can be part of the MEC server. In our case, we have shown that the neural network-based model can be placed on the MEC server and used for a variety of applications. Hence, the MEC server can be used to provide ultra-reliable low latency communications (URLLC) services that are part of one of the verticals in the 5G architecture.

The application discussed in this section is based on object detection via video streaming. The object detection model can be used to implement a variety of applications, like accident detection and vehicle detection in dense traffic environments in real-time scenarios. This paper limits the scope of application to the object detection model that identifies the object as a person or vehicle from the video.

The application is implemented in Python, using important libraries, including TensorFlow, OpenCV, etc. The end-user streams a video through an instance of a video analytic application set up at the MEC host (see Fig. 5 and Fig. 6). A client runs the video streaming application inside a UE Docker container, which is processed at the edge and sent to the cloud server based on the event trigger created at the edge application (operating at the MEC Host). Prior to MEC app instantiation, MEC Orchestrator registers the UE context creation request and provides a reference URL (MEC platform address) to the UE along with the context_Id and MEC app instance_Id. Based on the reference URL, the UE client app relays the video stream to the MEC platform. Different paths are set with the help of the IP route command to fulfil the integration of MEC with the 5G core setup.
Fig. 5 illustrates an example of *classifying objects* as individ-

uals at a place, whereas Fig. 6 depicts a *vehicle detection* use case on the road. The aforementioned illustrations are examples of object detection capability running inside the MEC platform over the proposed 5G testbed.

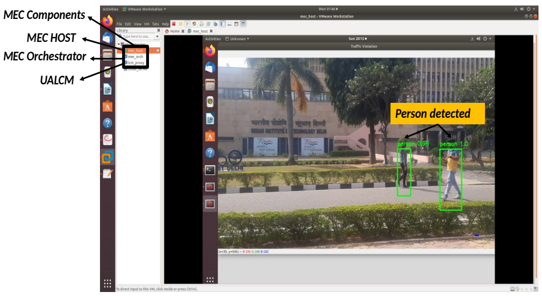

Fig. 5: Person detection algorithm in the MEC Host's platform.

It is important for the network provider to offer services that are customized to the requirements of the user. The previously mentioned services are possible with the help of network slicing provision in the 5G infrastructure. The proposed ETSI-compliant 5G testbed can be used for network slicing scenarios. On the UE side, based on the S-NSSAI parameter, different network slices can be allocated. Our 5G testbed provides slices for latency-sensitive applications such as traffic surveillance systems, haptic applications, etc. During the registration phase, the device application on the UE side transmits an S-NSSAI parameter for latency-critical services to the core network. The AMF then searches for available resources and allocates the optimal slice to the UE device application. The AMF, along with an SMF, will guide UPF to use the MEC host, where application-specific algorithms can be implemented for the aforementioned device applications. The above slice selection will reduce latency by implementing the algorithm on the MEC host instead of the cloud server.

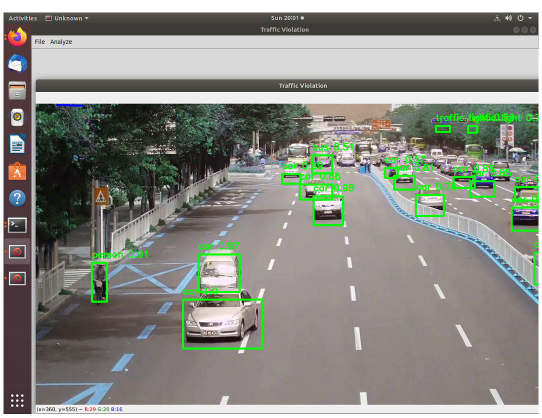

Fig. 6: Vehicle detection algorithm in the MEC Host's platform.

V. Conclusion

This paper proposed an open-source 5G MEC-OAI testbed, with the MEC being ETSI-compliant. A discussion of the existing standards was followed by the experimental setup. The objective is to support real-time, low-latency applications in a reliable manner via an open-source testbed that is easily deployable and hardware-independent. In order to validate the experimental setup, a video streaming application has been deployed on the testbed, addressing object detection use cases. As stated, the edge capabilities can be enhanced with the integration of network-slicing technology.

A single UPF setup, with the UPF being close to the edge, can aid achieving low latency, effectively meeting the prime objective of MEC. This can be taken up for further study.

References

[1] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin *et al.*, "Mec in 5g networks," *ETSI white paper*, vol. 28, no. 28, pp. 1–28, 2018.

[2] F. Giust, X. Costa-Perez, and A. Reznik, "Multi-access edge computing: An overview of etsi mec isg," *IEEE 5G Tech Focus*, vol. 1, no. 4, p. 4, 2017.

[3] 3GPP, "Architecture for enabling Edge Applications," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.558, 05 2022, version 17.3.0. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123558/17.03.00_60/ts_123558v170300p.pdf

[4] ETSI, "MEC Support for Network Slicing," ETSI, Group Report (GR) 024, 11 2022, version 2.1.1. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/MEC/001_099/024/02.01.01_60/gr_mec024v020101p.pdf

[5] A. A. Kherani, G. Shukla, S. Sanadhya, N. Vasudev, M. Ahmed, A. S. Patel, R. Mehrotra, B. Lall, H. Saran, M. Vutukuru *et al.*, "Development of mec system for indigenous 5g test-bed," in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2021, pp. 131–133.

[6] L. Cominardi, T. Deiss, M. Filippou, V. Sciancalepore, F. Giust, and D. Sabella, "Mec support for network slicing: Status and limitations from a standardization viewpoint," *IEEE Communications Standards Magazine*, vol. 4, no. 2, pp. 22–30, 2020.

[7] R. Pepito and A. Dutta, "Open source 5g security testbed for edge computing," in *2021 IEEE 4th 5G World Forum (5GWF)*. IEEE, 2021, pp. 388–393.

[8] M. Gharba, X. Xiao, H. Cao, J. Eichinger, A. Hecker, M. Kranzfelder, and D. Ostler, "A field test of 5g enhanced mobile ultrasound with network slicing," in *2021 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2021, pp. 1–6.

[9] F. A. Wiranata, W. Shalannanda, R. Mulyawan, and T. Adiono, "Automation of virtualized 5g infrastructure using mosaic 5g operator over kubernetes supporting network slicing," in *2020 14th International Conference on Telecommunication Systems, Services, and Applications (TSSA*. IEEE, 2020, pp. 1–5.

[10] M. Etsi, "Multi-access edge computing (mec); framework and reference architecture," *ETSI GS MEC*, vol. 3, p. V2, 2019.

[11] G. T. 23.401, "General packet radio service (gprs) enhancements for evolved universal terrestrial radio access network (e-utran) access," 2013.

[12] 3GPP, "System architecture for the 5g system (5gs); stage 2 (release 17): Ts 23.501," 2021.

[13] "Management and orchestration; Edge Computing Management," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 28.538, 05 2022, version 17.0.0. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/128500_128599/128538/17.00.00_60/ts_128538v170000p.pdf

[14] N. Sprecher *et al.*, "Harmonizing standards for edge computing-a synergized architecture leveraging etsi isg mec and 3gpp specifications," *ETSI White paper*, no. 36, 2020.

[15] 3GPP, "System architecture for the 5G System (5GS)," 3rd Generation Partnership Project (3GPP), Tech. Rep. 23.501, 10 2020, version 16.6.0. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf

[16] open air interface, "5g software alliance for democratising wireless innovation," 2014.