Sensor Virtualization and Data Orchestration in Internet of Vehicles (IoV)

Afaq Muhammad, Muhammad Saqib, Wang-Cheol Song*
afaq@jejunu.ac.kr, em.saqb@gmail.com, kingiron@gmail.com
Department of Computer Engineering, Jeju National University, Jeju-si, South Korea
*Corresponding author

***Abstract*—The idea of the Internet of Vehicles (IoV) has begun with the emergence of the Internet of things (IoT) and big data. This concept of IoV allows for information exchange between vehicles-to-anything (V2X), such as other vehicles, pedestrians, and mobile devices to enhance road safety and traffic management. However, network operation and management become more complex with an increase in the number of IoV devices. Although virtualization is a key enabler to providing flexible and automated network management and network resources optimization for vehicular services, a continuous information gathering from roadside infrastructure is greatly required. To this end, in this paper, a system is proposed to exploit the potential of Mobile Edge Computing (MEC) and virtualization technologies to maintain the status of the underlying network on the intermediate layer. This would lead to enhanced data processing performance as well as the availability of pre-processed or real-time data of the vehicular network to the newly introduced services despite the disconnection periods of vehicles.**

***Index Terms*—Internet of Vehicles, Sensor virtualization, OBU, MEC, 5G**

I. Introduction

The concept of intelligent transportation system (ITS) was put forth in the 90s. It involved application systems, such as traffic signal control, license plate recognizing, and vehicle management. For example, to extract and utilize dynamic and static information of vehicles, electronic tags installed on them could be identified by technologies like RFID. With a rapid development of wireless mobile communication technology, researchers started taking interest in communication between vehicles in order to enhance transportation efficiency and road safety. Vehicular Ad hoc Network (VANET) had been under the limelight for a long time. However, due to high-speed mobility of vehicles, the unreliability of service connection in VANET has been still a problem.

Consequently, requirements for future autonomous driving scenarios could not be met alone by VANET, since over the past few years, the road safety and traffic efficiency have been significantly affected by the increasing number of vehicles [1], resulting in a wider concern for the research and development of the related fields of Internet of Vehicles (IoV). By integrating the technologies of vehicle sensor data, data mining and automatic control, the IoV has been realized with intelligent traffic management and dynamic information service.

The applications of IoV can be categorized into three types, I.e., information services for a better user experience, driving intelligence for safety and efficiency, and intelligent transportation system for a smarter and more coordinated use of transport networks. Therefore, research on key technologies of such applications form the basis of 5G-IoV architectural design that is of great interest to a highly active research community. Since a variety of 5G entities such as 5G vehicle units, MEC Base Station (BS), and cloud servers are involved in 5G-IoV, the architecture is more flexible than the conventional IoV. Therefore, the V2X integration needs 5G connections unavoidably [2]. In addition to the V2X information interaction [3], 5G-IoV will also realize the interconnection of the On-Boarded Units (OBUs), MEC and cloud servers to support them with distinctive features and communication modes.

The OBU unit is typically a network device embedded in a roaming vehicle and has evolved from telematics services to a generic network device. This vehicle-mounted unit is used to maintain vehicle connectivity with the infrastructure or other vehicles using wireless communication modules. More significantly, it is likewise able to monitor and cache the in-vehicle sensor data, which is highly demanding in the field of urban mobility.

Connected vehicles is one of the most demanding use cases for 5G-IoV, which monitors the vehicle status information, to be used by various vehicular services. According to a demonstration [4], status information of the on-board sensors, neighboring vehicles data, and perceived location of the vehicles provide support to derive optimal driving decisions. Based on the automotive data, the opportunities for developing services are endless, ranging from smart cities, traffic management and so on:

1) Smart cities: vehicular data incorporation into a smart city application is useful for optimal route planning and flow optimization, congestion management and parking management, which can lead to efficient transportation management.

2) Traffic management: improves real-time traffic flows, efficient traffic light controls, public transport, and dynamic signage.

3) Safety emergency solutions: the data such as airbag triggering, hard braking, speed, and location give emergency responders a head start and identify problem areas to transportation planners.

4) Mapping planning solutions: Aggregated and blurred data generated by vehicles offer much deeper insight than simply tracking traffic speed. It helps in capturing location and

978-3-903176-32-4 © 2021 IFIP

environment information from connected vehicles to efficiently develop and update high-quality maps. It also helps to identify the detailed movement of vehicles and calculate the amount of time that the driver and passenger remain in an area to make smart decisions about opening or relocating commercial sites.

5) Remote diagnostics: real-time engine parameters such as oil temperature or tire pressure are used to monitor the health of the entire fleet. Hence, this could track a range of conditions that could make a vehicle out of service.

6) Predictive maintenance: real-time monitoring data that helps to predict the fault or need of maintenance.

Future vehicular services, especially in the field of smart city use cases, demand continuous information gathering with efficient processing capability. The current architectures in vehicular networks are covered in two a tier approach where the OBU-collected data is directly transmitted to the cloud servers for further processing. An intermediate processing and caching layer would be highly valuable for offloaded data processing nearby roadside infrastructure and the updated status information will always be accessible by the corresponding vehicular services running on the cloud platform, despite the disconnection periods of vehicles.

With the motivation of endless opportunities for the developing services, we exploit the potential of virtualization technologies to virtualize the vehicle-mounted terminals on the edge layer to offload the computational task processing from resources-constrained terminals and to make data accessible for various network services. By doing so, the roadside infrastructure data will always be accessible by the cloud services despite the disconnection periods. Also, it will save the radio resources and improve the data processing performance. On the MEC layer, we employ a pool of Virtual Network Functions (VNFs) to keep the corresponding vehicle terminal data, dynamically managed by Management and Orchestration (MANO) platform running on the cloud. The vehicular data may be further used by various cloud-operated services such as monitoring, analytic and prediction. The monitoring service maintains the updated information of all the VNFs running on the MEC and feeds the useful information to the corresponding modules. For example, a location prediction module is used to predict the vehicles' location at the next timestamp [5] and proactively instantiate the VNF instances on the corresponding MEC server to hide the instantiating time. Furthermore, we are using a simulated OBU called 'ELM 327OBD Simulator' which periodically generates sensory data for the system validation purpose.

The main contributions of this work can be summarized as follows:

- We propose a three-tier architecture by introducing an intermediate processing and caching layer for vehicle mounted OBU data.
- We exploit the potential of virtualization technologies to virtualize the vehicle-mounted terminals on the edge layer having priority for low latency actions, resulting in offloading computation-intensive tasks from vehicle-mounted terminals to the edge layer.
- For the system validation, we exploit the potential of an open source simulated OBU, which opens new opportunities for researchers to incorporate various use cases.

This article is organized as follows: Section 2 presents the related work. Section 3 presents the proposed three tier architecture to place the vehicular generated data on the intermediate (edge) layer. Section 4 presents the system testbed setup and validation. Finally, Section 5 concludes this article.

II. RELATED WORK

Network virtualization has been largely used in the data centers or other static network applications. Recently, some efforts have been made to replicate this success into dynamic networks where the location of nodes do not remain fixed such as vehicular and flying networks [6]. Virtualization approaches have been used in the vehicular network to maximize the utilization of the resources, improve the performance of context awareness and perception [7].

The focus of researchers has been shifted from cloud resources to vehicle resources virtualization i.e., to involve the vehicle mounted terminal or infrastructure. For example, after a successful deployment of virtualization into desktops and servers, an effort has been made to reflect this success into embedded applications [8]. The authors took advantage of the isolated running environment provided by virtualization technology and restricted the access to physical hardware which can lead to improved security of vehicular mounted units. Moreover, modern cars are equipped with intelligent on-board systems (IOSs) that provide support for multi-interconnection among vehicles and to the remote services, hence this can provide the surrounding information to the vehicles and increase road safety. An associated problem is that the closed loop architecture in such built-in systems does not support the upgradation feature. With this realization, the concept of virtualization technology has been replicated into vehicles to execute the vehicular services as software applications on standard IT platforms. By doing so, the closed IOSs can be transformed to open ones that can be upgraded in a cost-effective manner [9]. In addition, there is a special need to coordinate the use of resources (i.e., sensing, processing, and storage capabilities) offered by field area networks. A complete virtualization approach has been used to virtualize the embedded resources of vehicular terminals and deploy different tenant services on the same underlying mobile substrate, also enabling the migration feature of virtual machines between OBUs [10].

On the OBU side, research has also been done on the usage of virtualization. In [11], the authors introduced the concept of enhancing vehicular units by employing virtualization and managing the access to physical hardware. A more sophisticated approach is the utilization of NFV in the OBU to construct an architecture that is flexible and that could be managed and updated remotely [9]. In [10], authors have considered migration approaches of virtual machines (VMs) between OBUs to provide services based on the available resources.

MEC-based design for a vehicular network is a very recent area of study which provides a range of benefits from high computational processing to low latency services [12]. In [16], authors presented an MEC layer employed in roadside infrastructure. MEC-assisted network architecture is a promising solution to improve the efficiency and flexibility of vehicular networks [13]. It has also been used as a potential approach for ultra-low latency communication in a highly dense vehicular network [14]. A more advanced approach has been used to exploit the capability of the MEC to ensure data communication reliability of vehicles under time varying wireless channel constraint [15].

An architecture is presented for 5G-enabled use cases, which incorporates MEC and cloud services to improve road safety and enables autonomous driving with a special focus on time critical services, but the flexibility of NFV was not exploited [16]. Another work proposes a 5G platform which considers NFV and orchestration [17]. Similarly, one more 5G-enabled platform uses VNFs on the MEC layer for the sake of status information collection from roadside infrastructure to detect potential collisions [18]. Although the authors developed a reference 5G platform for vehicular application, it is not open for further intelligent transportation services (ITS) or connected vehicles use cases.

So far, there is no single approach that focuses on sensor virtualization and data orchestration in vehicular networks, to free up the resource-constrained vehicle mounted terminals and make the network status information available to the developing vehicular services. Our novel approach exploits the virtualization capabilities of 5G architecture by placing the vehicular unit's data on the edge layer, hence lies between roadside infrastructure and the cloud.

III. Design and Deployment

As illustrated in Fig. 1, the proposed network model is segregated into three tiers; each tier includes different components. Independent functionalities are associated with each component, working in a cooperative way. Below are the details of the functionalities of each component belonging to different tiers.

Tier 1: The first tier consists of monitoring, prediction, and orchestration services. Monitoring service plays an enabling role for prediction and orchestration services, providing the updated state information of the underlying network infrastructure. Since the nature of vehicular networks is very dynamic, the network topology does not remain stable. In addition, the coverage distance of MEC/BS is limited, vehicles may traverse multiple MEC platforms. For that reason, service continuity is solely needed to provide an acceptable level of service while a vehicle passes over different network domains. The only possible solution is to predict and calculate the vehicular node location at the next time stamp based on GPS data. Thus, a location prediction module has been used to calculate the number of vehicles that might traverse multiple MECs over a certain period. The MANO entity is employed to orchestrate the on-demand resources and services or the underlying network. The MANO takes advantage of future location prediction service and proactively instantiates or migrates the instances between MEC platforms, which can resolve the scalability issue by hiding the instantiating time, the amount to register a vehicle to the virtual counterpart. In addition, MEC application manager handles different services required by a vehicle in accordance with different types of subscription models. Network service designer module generates a generic design of virtual network function forwarding graphs (VNFFG) for the required MEC service. The policy configurator module generates the orchestration policies and VNF instantiation and resource configurations for implementation on the physical layer. The proposed system follows a closed-loop control operation that monitors all the vehicles data on runtime, updates the service parameter, and handles the fast mobility of the vehicles. Moreover, for the visualization of the monitored data, Grafana monitoring tool has been deployed in the testbed that allows to query, visualize, and generate alerts.

Tier 2: The MEC layer hosts a separate virtual instance for each physical vehicle in the form of VNFs. The virtual instance stores vehicle status information inside a real-time database called RethinkDB [19]. This virtualization provides an abstracted view of OBUs to the middle and upper tiers. All the instances share their updated information with the monitoring service running on the upper tier. Moreover, it also serves the data requests from the cloud services and gets on-demand data from the bottom layer entities.

Tier 3: This layer consists of vehicles in a simulated environment, which periodically generate data attributes that specify location, engine status, speed, and so on. These parameters are generated from the vehicle Electronic Control Units (ECUs), Controller Access Networks (CANs), and even infotainment systems. OBD2 is a higher layer protocol, while CAN bus is a method for communication. OBD2 is used to read the ECU data using CAN bus and supports a range of standard parameters IDs (PIDs) that can be logged across most of the modern vehicles. To be more specific, we can easily get human-readable data from vehicles such as speed, RPM, engine status etc.

Initially, the simulated OBU triggers a request to the instance (OBU) manager, located at tier 1, to establish its connectivity to the virtual counterpart. The instance manager running as a VNF, handles mapping between physical and virtual counterparts. The first allocation request then goes to the MANO entity, which instantiates the corresponding instance on the MEC layer and notifies the instance manager in return. After the instantiating process, the simulated OBU starts data sharing with the corresponding instance located in tier 2.

Data requests from services can be determined at three tiers in the proposed system. When data is required by service VNF from tier 1, it initiates a data request message that is handled by the data analytics module. If the data had not been previously acquired, the particular vOBU is examined by the vOBU manager and the request is sent to it.

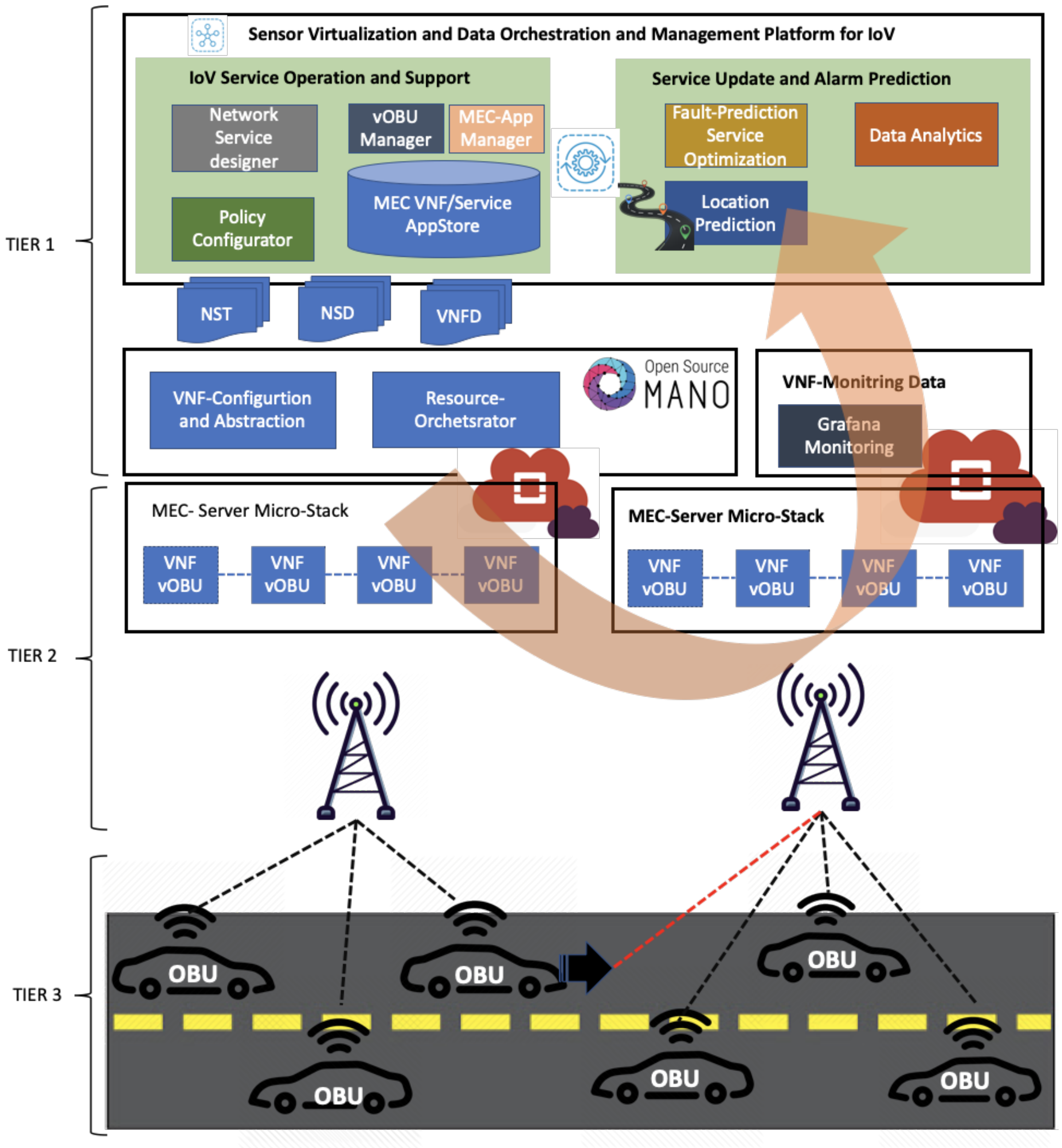

Fig. 1. Proposed system architecture

IV. Result and Discussions

The testbed used to validate the proposed system consists of a virtualization infrastructure that runs on two different Lenovo servers dedicated to MEC and Cloud, respectively. Both servers are with specifications of 16 GB RAM, 500 HDD and Intel (R) Core (TM) i9-900 CPU at 3.10 GHz, run Ubuntu20.4 as a host operating system. In addition, OpenStack has been used as a cloud operating system as it allows organizations to create a completely custom cloud that is compatible with OSM MANO and caters to domain specific needs.

All the modules and services are running as VNFs using an Ubuntu 18.04 operating system (CLI), on top of the OpenStack. With the realization of software configurations and updates, the base images have been prepared with all the pre-requisites, to be instantiated by MANO onto the OpenStack. For example, a real-time database package has been installed with all the prerequisites on the vOBU image, which uses real-time processing to handle the workloads whose state is constantly changing. In total, five types of VNFs have been on-boarded to the MANO platform with various specifications as shown in Table 1.

As illustrated in Fig. 2, instances have been instantiated on to the MEC platform by MANO. Each instance maintains its associated vehicle data into the real-time database named RealthinkDB and shares the state information to the monitoring service deployed in the data center (cloud). A customized web page is designed for monitoring service to show several

TABLE I
ONBOARDED VNF SPECIFICATIONS

VNF Name	Specifications		
	CPUs	***RAM(GB)***	***HDD(GB)***
Virtual OBU	1	1	4
OBU Manager	2	2	4
Monitoring Service	2	2	50
Location Prediction Service	2	2	50
Data Analytics	2	2	50

vehicles parameters using plots. The data analytic module keeps the pre-processed information, which can be later used by various network services.

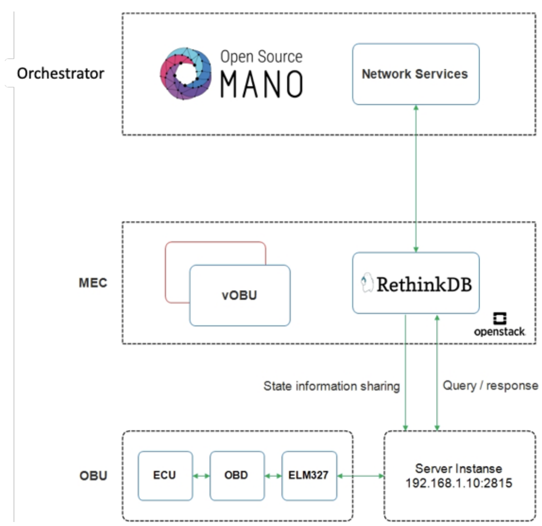

Fig. 2. Methodology for the working of vOBU instantiation and data management

In addition, we used an OBD2-compliant simulator named ELM327, adhering to the SAE standard J/1979 and ELM327 command protocol. ELM327 OBD Simulator creates a server instance at the specified address and port that accepts query bytes, processes them, and returns a response. To get the engine sensor specific data, two bytes are written to the OBD2 port (query bytes), which return up to eight bytes in response. OBD2 compliant formulas are then used to calculate the actual value from the bytes.

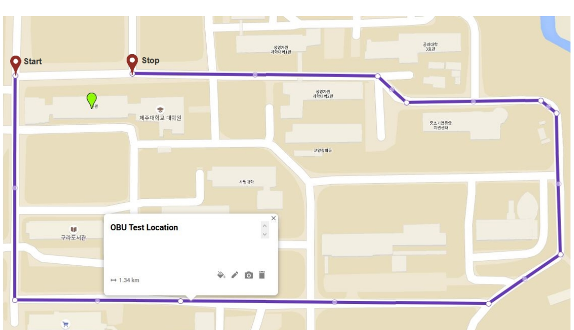

Fig. 3. OBU trial map

ELM327 supports up to 60 parameters that can be logged across most of the modern cars. We used a total of four parameters including engine coolant temperature, engine speed, vehicle speed and ambient air temperature. Moreover, as OBU in the simulated environment is not moving, but a static entity. Therefore, we have generated GPS coordinates using GPS Google map for each OBU entity separately and loaded these coordinates to the simulator. As shown in Fig. 3, the test location has been set to the area of Jeju National University, South Korea. We assume that the vehicle starts moving from a starting point and stops at an ending point on the map. The total length of the circular paths is 1.34 km. During the trial, the variation in the engine specific parameters was periodically reported to the corresponding instance in the virtual counterpart, which was further accessed by monitoring service. This data can then be viewed in the form of plots on a customized web page, and further analyzed using ML techniques.

The proposed system operation has been successfully validated by accessing the monitoring service data from Grafana monitoring tool. The resulting values for four different parameters are reflected in Fig. 4. The data showed are the speed, revolutions per minute (RPM), coolant engine temperature, and ambient air temperature. As illustrated, there is a positive correlation between vehicle speed and RPM while the coolant engine temperature slowly increased as the vehicle moved on the plotted area over the map. The ambient air temperature almost remained the same over total trail time. So, the variation in the engine parameters validate the successful operation of our proposed system, as roadside data has been accurately recorded at the intermediate layer. Moreover, there are up to 60 engine parameters in our simulated OBU but the most common four parameters have been selected for the trial to verify the system operation while the rest of the parameters can be used by various use cases in the future.

V. CONCLUSION

The 5G and beyond technologies will revolutionize the transportation system by introducing new services to the area of safety, traffic management and more. The proposed work provides a roadmap to virtualize the vehicular network data by exploiting the virtualization and edge computing capabilities. By doing so, newly introduced services can incorporate the surrounding information with the vehicles' generated data and can accomplish the vision of connected vehicles in a true way. This has been the first effort to enable sensor virtualization and data orchestration in vehicular networks. To this purpose, a three-layered system has been proposed, which includes MEC, cloud services, and leverages the potential of virtualization technologies to virtualize the OBUs on the edge layer. More importantly, a pool of VNFs has been deployed to manage the vOBU data with MANO platform. Furthermore, basic successful demonstration of the proposed system has been shown to validate the system operations by retrieving the monitoring service data from Grafana monitoring tool. A variety of engine-specific parameters have been periodically

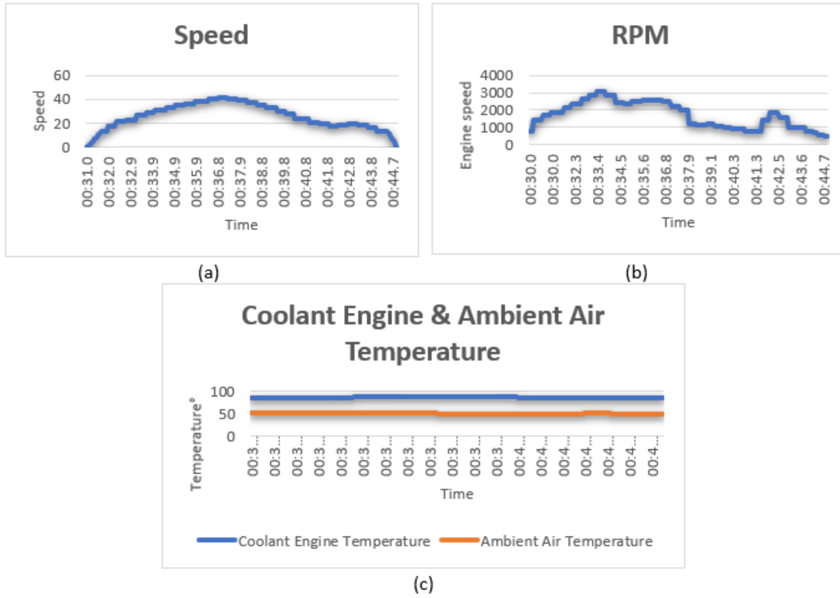

Fig. 4. Monitoring service results; (a) vehicle speed, (b) engine rpm, (c) engine coolant and air temperature

reported to vOBU at the MEC layer. The monitoring service in the proposed system enables to view this data in the form of plots and makes it available to tier 1 for further analysis, such as location prediction and fault prediction service.

ACKNOWLEDGMENT

This research was supported by the Korea Research Fellowship program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (2019H1D3A1A01102980). This research was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B01016322).

REFERENCES

[1] H. A. Omar, N. Lu, and W. Zhuang, “Wireless access technologies for vehicular network safety applications,” *IEEE Network*, vol. 30, no. 4, pp. 22–26, 2016.
[2] C. R. Storck and F. Duarte-Figueiredo, “A 5g v2x ecosystem providing internet of vehicles,” *Sensors*, vol. 19, no. 3, p. 550, 2019.
[3] B. Di, L. Song, Y. Li, and Z. Han, “V2x meets noma: Non-orthogonal multiple access for 5g-enabled vehicular networks,” *IEEE Wireless Communications*, vol. 24, no. 6, pp. 14–21, 2017.
[4] S. Pandit, F. H. Fitzek, and S. Redana, “Demonstration of 5g connected cars,” in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2017, pp. 605–606.
[5] M. T. Abbas, M. A. Jibran, M. Afaq, and W.-C. Song, “An adaptive approach to vehicle trajectory prediction using multimodel kalman filter,” *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 5, p. e3734, 2020.
[6] N. Pathak, S. Misra, A. Mukherjee, A. Roy, and A. Zomaya, “Uav virtualization for enabling heterogeneous and persistent uav-as-a-service,” *IEEE Transactions on Vehicular Technology*, 2020.
[7] M. R. Jabbarpour, A. Marefat, A. Jalooli, and H. Zarrabi, “Could-based vehicular networks: a taxonomy, survey, and conceptual hybrid architecture,” *Wireless Networks*, vol. 25, no. 1, pp. 335–354, 2019.
[8] J. Pelzl, M. Wolf, and T. Wollinger, “Virtualization technologies for cars: Solutions to increase safety and security of vehicular ecus,” in *Proceedings, embedded world Conference, Nuremberg, Germany*, 2009.
[9] M. Zhu, J. Cao, Z. Cai, Z. He, and M. Xu, “Providing flexible services for heterogeneous vehicles: an nfv-based approach,” *IEEE network*, vol. 30, no. 3, pp. 64–71, 2016.
[10] B. Baron, M. Campista, P. Spathis, L. H. M. Costa, M. D. de Amorim, O. C. M. Duarte, G. Pujolle, and Y. Viniotis, “Virtualizing vehicular node resources: Feasibility study of virtual machine migration,” *Vehicular Communications*, vol. 4, pp. 39–46, 2016.
[11] J. Pelzl, M. Wolf, and T. Wollinger, “Virtualization technologies for cars, solutions to increase safety and security of vehicular ecus,” *Automotive–Safety & Security, Stuttgart, Germany*, 2008.
[12] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, “Vehicular edge computing and networking: A survey,” *Mobile Networks and Applications*, pp. 1–24, 2020.
[13] H. Zhang, Z. Wang, and K. Liu, “V2x offloading and resource allocation in sdn-assisted mec-based vehicular networks,” *China Communications*, vol. 17, no. 5, pp. 266–283, 2020.
[14] S. Zhou, P. P. Netalkar, Y. Chang, Y. Xu, and J. Chao, “The mec-based architecture design for low-latency and fast hand-off vehicular networking,” in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, pp. 1–7.
[15] H. Ke, J. Wang, L. Deng, Y. Ge, and H. Wang, “Deep reinforcement learning-based adaptive computation offloading for mec in heterogeneous vehicular networks,” *IEEE Transactions on Vehicular Technology*, 2020.
[16] T. Ojanperä, J. Mäkelä, O. Mämmelä, M. Majanen, and O. Martikainen, “Use cases and communications architecture for 5g-enabled road safety services,” in *2018 European Conference on Networks and Communications (EuCNC)*. IEEE, 2018, pp. 335–340.
[17] D. Rapone, R. Quasso, S. B. Chundrigar, S. T. Talat, L. Cominardi, A. De la Oliva, P.-H. Kuo, A. Mourad, A. Colazzo, G. Parmeggiani *et al.*, “An integrated, virtualized joint edge and fog computing system with multi-rat convergence,” in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 2018, pp. 1–5.
[18] N. Sambo, L. Valcarenghi, P. Iovanna, G. Imbarlina, F. Ubaldi, T. Pepe, A. Garcia-Saavedra, G. Landi, I. Pascual, R. Martínez *et al.*, “Mobile transport and computing platform for 5g verticals: Resource abstraction and implementation,” in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2018, pp. 1–6.
[19] G. Tiepolo, *Getting started with rethinkdb*. Packt Publishing Ltd, 2016.