A Comparison of Communication Mechanisms in Vehicular Edge Computing

Liangkai Liu
Wayne State University

Baofu Wu
Wayne State University

Weisong Shi
Wayne State University

Abstract

Connected and Autonomous Vehicles (CAVs) is becoming more and more popular for automobile academic and industry community. Communication mechanisms play an important role in CAVs applications and services. However, lack of detailed comparison of different communication mechanisms is the main obstacle for the deployment of CAVs applications and services. In this paper, we set up an end-to-end prototype which supports WiFi, LTE, and DSRC based communications and evaluate the performance in latency, power dissipation, and system utilization. Three observations are summarized for the real deployment of VEC applications.

1 Introduction

Due to the safety and efficiency in fuel consumption, autonomous driving techniques have attracted huge attention from both the academic and industry [14]. According to [2], the global autonomous driving market is expected grow up to $173.15B by 2030. However, there are still several challenges in the development and deployment of the autonomous driving system. The first challenge is on the cost. According to [1,18], the cost of a level 4 autonomous driving vehicle can attain 300,000 dollars, in which the sensors and computing platform cost almost 200,000 dollars. The second challenge is the real-time requirements of the sensing, perception, and decision. According to [9], when the vehicle drives at 40 km per hour in urban areas and that autonomous functions should be effective every 1 meter, the execution of each real-time task should be less than 100ms. However, limited by the performance of perception algorithms and the stochastic run time on the computing platform, how to guarantee the real-time requirement of the safety-critical system is still an open question. The third challenge is the reliability of cameras and LiDAR under severe weather conditions. The quality of images captured by cameras is affected by the lighting conditions [13] while the point clouds from LiDAR can be affected by noisy and sparse.
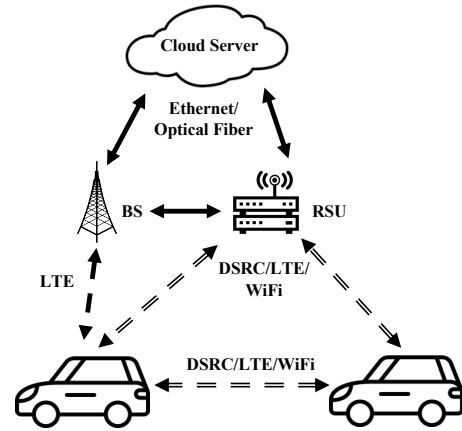

Figure 1: A typical example of vehicular edge computing.

In order to address the above challenges, vehicle-to-everything (V2X) based vehicle edge computing attracts more and more attention. V2X enables the vehicle to obtain real-time traffic information from the Road Side Unit (RSU), which helps to decrease the cost of sensors on an autonomous driving vehicle. Meanwhile, there would be less sensing and perception tasks with real-time requirements and the reliability of cameras and LiDAR would not be a issue any more. Building intelligent traffic infrastructures with complex computation and communication resources become a consensus of the transportation community for the future transportation system.

For the state-of-the-art of VEC, lots of works are focused on task offloading algorithms to optimize the usage of the V2X communications resources. In [6], a joint offloading decision and resource allocation model is designed to optimize the performance of task offloading when the computation requirement is unknown. Similarly, [19] proposes a service caching and task offloading model to optimize the cost of the computation while [5] considers the offloading problems for the energy efficiency. However, all these works are based on simulations and the evaluation in real environment is missing.

Besides, for the edge computing enabled application prototypes discussed in [12, 20], the communication mechanisms are limited to WiFi. Other potential communication mechanisms like LTE and DSRC are not touched. A general and comprehensive comparison of different types of V2X communication mechanisms for VEC applications is still missing.

In this paper, we set up a VEC prototype which supports three typical V2X communications including WiFi, Long-Term Evolution (LTE), and Dedicated Short Range Communication (DSRC). We set up dedicated end-to-end WiFi communication through a router, an LTE communication using Software Defined Radio (SDR) device USRP B210 boards with wireless antennas, and a DSRC communication using commercial products with On-Board Unit (OBU) and Road-Side Unit (RSU). On top of the prototype, several applications including *Robot Operating System (ROS)*, *Socket*, and *Ping* messages are implemented to evaluate the performance of the communication as well as their impacts to the computation [4, 16]. The evaluation covers the round-trip latency, system utilization, and power dissipation. We summarize several observations towards the usability, availability, and efficiency of the V2X communications to the VEC applications. The contributions of this paper are as follows:

- To the best of our knowledge, our work is the first comparison of communication mechanisms of LTE, WiFi, and DSRC using real VEC applications.

- An end-to-end communication prototype is built which can support LTE, DSRC, and WiFi. On top of the prototype, *ROS*, *Socket*, and *Ping* messages are implemented.

- We evaluated the communication prototype in latency, power dissipation, and system utilization and get three observations for the real deployment of VEC applications.

The remainder of paper is organized as follows. Sections 2 and 3 depict the motivation and methodology for the communication-computation comparison, respectively. We present the comparison results in Section 4, and summarize the paper in the last section.

2 Background and Motivation

Considering a typical VEC scenario showing in Figure 1, the cloud, base station (BS), RSU, and vehicles are connected through a three-tier communication network. The bottom tier is for vehicles, where vehicles can share information through DSRC, LTE, as well as WiFi. The middle tier includes the BS and RSU. Vehicles communicate with the RSU through LTE, DSRC, and LTE while they can also communicate with the BS through LTE. The top tier is the cloud data center. Both BS and RSU are connected to the cloud through Ethernet or optical fiber connections.

Table 5 shows a brief comparison of LTE, DSRC, and WiFi in terms of frequency and max range. In practical, DSRC is designed for sharing information between vehicles and traffic infrastructure. There are seven lists the fifteen message types that are defined in the SAE J2735 standard [17], which covers information like the vehicle's position, map information, emergence warning, etc [10]. Limited by the available bandwidth, DSRC messages have small size and low frequency. However, DSRC provides reliable communication even When the vehicle is driving 120 miles per hour. In contrast, WiFi and LTE provides more bandwidth but performs poor in mobile environment. The coverage is another concern when implementing the V2X application. Due to the mobile cellular network technology, LTE is the best in terms of coverage while DSRC also achieves almost 1000 meters' coverage with the speed at 120 mph. The coverage of WiFi shows different for indoor and outdoor scenarios. For 802.11ac, it achieves 46 meters for indoor scenario while 92 meters for outdoor.

3 Methodology

In order to compare the performance of different V2X communication mechanisms and evaluate their impact to the computation, we build an end-to-end prototype which supports WiFi, LTE, and DSRC. In this section, first we discuss their setups, then we discuss the communication messages defined for the comparison.

3.1 Hardware Description

For wifi-based communication, a TPlink router is used to set up a hotspots. The router supports two frequency band: 2.4GHz and 5GHz. For LTE-based communication, two Ettus Research USRP B210 boards with two VERT2450 antennas are used with *uhd* software driver to set up the network. For DSRC-based communication, we use moKar DSRC devices with two DSRC antennas in the prototype.

All the communications antennas/devices are connected to a host machine to handle the communication and application development. For the on-board computing boards, we use Intel Fog Reference and Nvidia Jetson TX2. The Intel Fog Reference has 8 Intel Xeon(R) CPU with 3.60GHz frequency. And it has 32GB memory. Jetson TX2 is a typical mobile computing board with GPU installed [21]. For RSU, a cluster of Intel Fog Reference boards are connected to provide computation as well as communication resources to vehicles [11]. NVIDIA® Jetson TX2 is a power-efficient embedded AI computing platform which has dual-core Denver and quad-core ARM® Cortex-A57 CPU equipped with 8GB DDR4 memory and 32GB eMMC. The GPU on Jetson TX2 is powered by NVIDIA Pascal™ architecture with 256 CUDA cores.

Four types of messages are leveraged to transmit in the end-to-end bi-direction prototype. We choose three ROS-based messages and one Internet Control Message Protocol (ICMP)
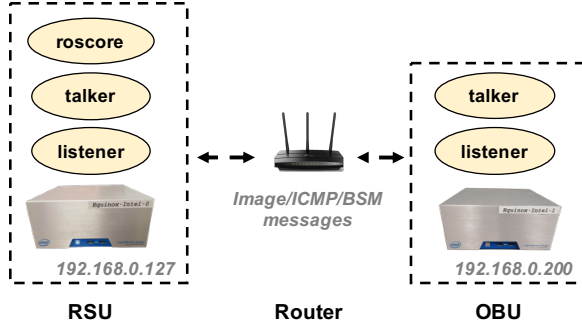
Figure 2: WiFi-based communication.

Figure 3: LTE-based communication.

based message: *BSM*, *image1*, *image2*, and *Ping*. Basic Safety Message (BSM) is one of the fifteen predefined messages to share the vehicle's position, speed, direction, with timestamp. We choose *BSM* and implement the message sharing through ROS. The size of *BSM* message is 1416 bytes. In addition, two image *ROS* messages are defined and they include the header which has the timestamp, size, an array to store image pixels, etc. We choose two images: one is 91kB, another is 401kB. *Ping* is another message we used for the evaluation of latency and power dissipation. The size of *Ping* message is 64 bytes. Since DSRC-based communication does not generate a virtual IP address and it does not support transmitting images, only the predefined BSM message is implemented. For WiFi and LTE, all four messages are implemented and evaluated.

3.2 WiFi-based Communication

As one of the most popular wireless communications for large volume of data transmission, WiFi is a potential communication methods for VEC applications. Figure 2 shows the communication framework implemented on top of WiFi. When OBU and RSU connected to the router, unique IP addresses are assigned and they are able to *Ping* each other within the local area network. In this paper, we use 5GHz WiFi connection.

On top of the WiFi connection, we implement a *ROS-based* communication for data sharing. ROS is a communication middleware designed for robots as well as connected and autonomous driving vehicles [3]. In ROS, a master is launched to manage all the ROS nodes, topics, services, and actions. *roscore* is used to launch the ROS master in RSU. Two ROS nodes are launched in both RSU and OBU: talker, listener. They are defined to send ROS messages between RSU and OBU through Inter-Process Communication (IPC).

3.3 LTE-based Communication

Compared with WiFi, LTE has good strength on the coverage of the communication and its performance in mobile environment. Driven by the development of Software De-
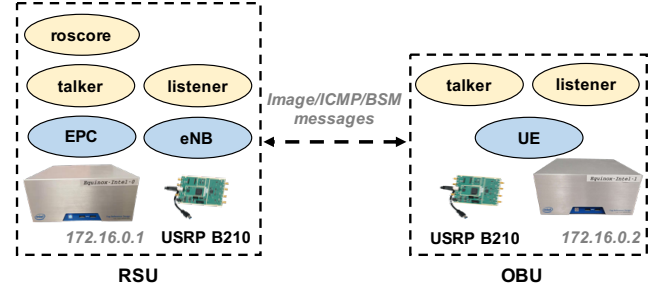
fined Radio (SDR), the programmability of the wireless communication devices improves a lot and it becomes easier to set up a dedicated LTE communication testbed using SDR boards with open sourced software like *OpenAirInterface* and *srsLTE* [7, 15].

In our prototype, the LTE communication is set up by using two Ettus Research USRP B210 boards with VERT2450 antennas [4]. USRP B210 boards are connected to the host machine through USB 3.0 interface. For the software, *uhd* is built as the USRP driver and *srsLTE* is installed to build the whole LTE network. *srsLTE* is an free and open-source 4G/LTE software suite which includes core network (EPC), base station (eNB), and user equipment (UE) [7]. All the modems can be implemented as an application running in a standard Linux-based machine.

As is shown in Figure 3, EPC and eNB are implemented on the RSU because they need to provide LTE connections to all nearby vehicles. UE is implemented on the OBU for registration and connection to the eNB. To begin with, EPC with eNB are launched to broadcast signals at a specific frequency. After UE is launched on another machine, it will search for the area to find the cellular tower and try to set up connection. When the UE is registered in EPC and connected with the eNB, an unique IP address will be assigned to the UE. For example, a virtual network interface is implement by eNB on the RSU (172.16.0.1) and it assigns 172.16.0.2 to the UE. With unique IP addresses in the local area network, RSU and OBU can ping each other and the networking programming interface can be used for application development. Similar to WiFi, the data sharing application is also implemented based on ROS. ROS master with talker and listener ROS nodes are implemented on the RSU while talker and listener ROS nodes are implemented on the OBU. We use the default frequency of *srsLTE* called EARFCN 3400 with single VERT 2450 antenna, which represents the download link with 2685MHz and upload link with 2565MHz.

3.4 DSRC-based Communication

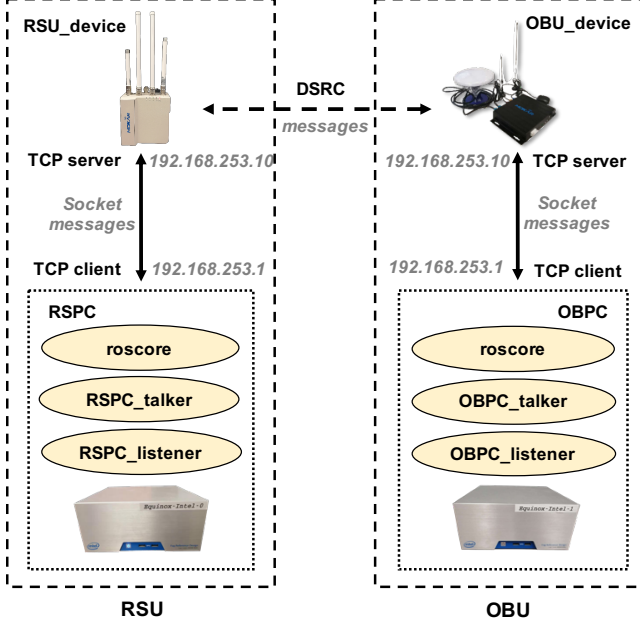DSRC-based Communication enables RSU and OBU to share short messages such as safety warnings and traffic information

Figure 4: DSRC-based communication.

Table 1: The end-to-end latency from RSU to OBU.

RSU -> OBU (ms)	LTE	WiFi	DSRC
BSM	**3.12**	67.96	8.46
image1	6174.32	**1733.90**	-
image2	20220.85	**1741.07**	-
Ping	**25.02**	50.49	-

Table 2: The end-to-end latency from OBU to RSU.

OBU -> RSU (ms)	LTE	WiFi	DSRC
BSM	51.02	14.90	**9.84**
image1	6012.8	**1301.38**	-
image2	21876.4	**1293.25**	-
Ping	**28.22**	67.52	-

Since time difference affects the latency results, we set up Network Time Protocol (NTP) based time synchronization and use *clockdiff* in Linux to measure the clock difference when transmitting messages. Then the clock difference is applied calculate the end-to-end latency. The results of the average end-to-end delay are shown in Table 1 and Table 2. From RSU to OBU, LTE performs the best for *BSM* and *Ping* while WiFi is better for two image messages. From OBU to RSU, DSRC is the best for *BSM* message. For LTE-based communication, we notice that LTE performs better than WiFi for *BSM* and *Ping* but worse for *image*. In addition, DSRC shows stable performance in transmitting small messages.

Besides the average end-to-end delay, we also draw the CDF of the end-to-end delay in two directions for transmitting *BSM* message, which are shown in Figure 5. We can find that WiFi shows large variance for RSU to OBU (over 300ms) while LTE shows large variance for OBU to RSU (almost 100ms). We think one reason for the variation of LTE is the bandwidth difference of downloading and uploading. The variance of DSRC is always the lowest which means the latency of DSRC is very stable.

In order to measure the latency in mobile environment, HydraOne is also used in the experiments to communicate with the RSU [21]. We choose *BSM* message and measure

with the communication frequency of 5.9GHz band. Our prototype of DSRC-based communication is showing in Figure 4. RSU and OBU can share messages with each other, while message sent from RSPC and OBPC can go through RSU_device and OBU_device. Host machines (RSPC, OBPC) and DSRC devices (RSU_device, OBU_device) are within the same LAN. DSRC communication is used between RSU_device and OBU_device.

For both of RSPC and OBPC, two ROS nodes named *talker* and *listener* are created. *talker* reports information of vessel, such as latitude, longitude, heading and speed at frequency of 250Hz while listener subscribes to it with the same frequency. TCP Socket-based communication is setup between host machines and DSRC devices. The RSU_device and OBU_device communicate with each other through DSRC to share BSM messages at 250Hz.

4 Comparison and Observation

In order to evaluate the performance of the end-to-end communication prototype, we conducted experiments and evaluated in three aspects: end-to-end latency, power dissipation, and system utilization.

4.1 End-to-end Latency

End-to-end latency is one of the most important metric to evaluate the performance of applications which heavily relies on the communication. In this part, latency is measured in two directions: from RSU to OBU and from OBU to RSU.
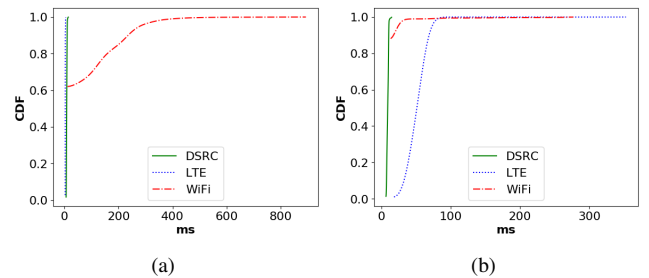

Figure 5: The CDF of *BSM* message end-to-end latency in: (a) RSU to OBU, (b) OBU to RSU.

Table 3: The latency of transmitting BSM message with different speed.

Latency (ms)	LTE	WiFi	DSRC
0 m/s	**3.12**	15.06	8.88
3 m/s	657.28	12.99	**9.12**
5 m/s	1071.52	16.18	**9.21**

Figure 6: Power Dissipation.

Table 4: The overhead of WiFi and LTE communications.

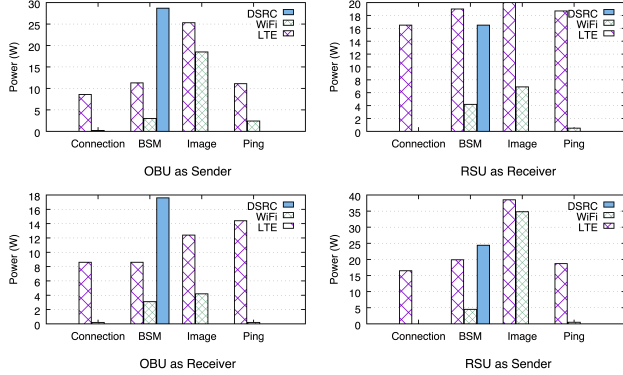| Metrics | RSU | | OBU | |
	roscore LTE	roscore WiFi	eNB	UE
memory (kb)	50364	55400	**1599532**	**1488000**
CPU (%)	2.22	1	**27.73**	**23.56**

the latency from RSU to OBU with LTE, WiFi, and DSRC. From the results shown in Table 3, DSRC shows the lowest difference when the speed of HydraOne changes, while LTE's latency increased by over 300 times when the speed increases from 0 to 5 m/s. For WiFi-based communication, negligible difference is observed when the speed changes. From the discussion above, we can generalize our first observation:

Observation 1: *DSRC shows fast and reliable performance for transmitting predefined messages. WiFi has the lowest latency when transmitting images but its variance for BSM and ping is large. LTE performs well for small messages like BSM and ping but bad for large messages like images. LTE and DSRC have stable performance under mobile environment while LTE has the most performance degradation. A communication mechanism which can provide sufficient and stable bandwidth is still missing.*

4.2 Power Dissipation

For the measurement of power dissipation, Watt's Up Pro is used to record the hardware power dissipation and *rapl* tool from Intel is used to measure the processor power dissipation [8]. Since both OBU and RSU can be sender and receiver, we measure the power dissipation for communication connection and message transmission in four cases, which are shown in Figure 6.

From the results, we can find that WiFi is always more energy efficient than LTE and DSRC. For the transmission of *BSM* message, DSRC has the most energy consumption for most cases and they are all larger than 16W. In addition to

the data transmission, communication setup/connection also consumes a lot of energy. The connection of LTE consumes 8.6W for OBU and 16.5W for RSU, which is the power overhead of setting up EPC and eNB on RSU and UE on the OBU. Beside, transmitting image messages consumes more energy than others. On sender's side, the power dissipation for image sending is larger than 18.5W. This fact leads to the following observation:

Observation 2: *There are non-negligible power dissipation for communication connections for LTE and DSRC. Since sending messages also consumes lot of energy, how to make the V2X communication more energy efficient is still an open question.*

4.3 System Utilization

For the measurement of system utilization, we use *top* to get the memory footprint and the CPU utilization. The results are shown in Tables 6, 7, 8, and 9 (in Section 5). It can be found that DSRC has the lowest memory footprint and CPU utilization for *BSM* message, while LTE has less memory footprint and CPU utilization when RSU is the receiver. For WiFi, the memory footprint and CPU utilization is higher than others except when RSU is sending image message.

In addition to the system utilization for sending messages, there are also some overhead in setting up the communications. On RSU, the extra overhead includes launching ROS master node through *roscore* and launching the EPC with eNB for LTE communications. On OBU, UE needs to be launched for LTE connection. The results for the above overhead are shown in Table 4. We can find that ROS master does not consume a lot memory footprint but eNB and UE consume large memory footprint and CPU resources. This fact leads to the following observation:

Observation 3: *Sending process consumes more memory footprint and CPU resources than receiving. The usage of srsLTE as well as ROS introduces non-negligible system overhead.*

5 Summary

In this paper we compared three communication mechanisms for VEC indoor and summarized three observations for developing real VEC applications.

Discussion Topics

What kind of feedback you are looking to receive

We are looking to receive the feedback on the potential optimizations that can address the inefficiencies on latency, power dissipation, and system utilization observed in our experiments.

The controversial points of the paper

Currently all the experiments are conducted indoor, which may have some difference with the outdoor scenario.

The type of discussion this paper is likely to generate in a workshop format

How to leverage the V2X communications for vehicular edge computing?

The open issues the paper does not address

The scenario of high speed (>20m/s) and longer distance are not evaluated in this paper. Also, C-V2X-based communication is not included in the prototype.

Under what circumstances the whole idea might fall apart

The idea might fall apart if we consider the outdoor case with speed larger than 20m/s or distance longer than 1000 meters.

Appendix

Table 5: A comparison of V2X communications.

Communications	Bandwidth	Max Range
Wi-Fi (802.11ac)	2.4GHz / 5GHz	Indoor: 46m Outdoor: 92m
4G/LTE	67MHz	—
DSRC	75MHz	1000m (120mph)

Table 6: The memory footprint of RSU as sender and receiver.

RSU (kb)	msg	LTE	WiFi	DSRC
sender	*BSM*	49960	50084	**19700**
	image	264872	**264700**	-
receiver	*BSM*	50364	264872	**1588**
	image	**120808**	125196	-

Table 7: The CPU utilization of RSU as sender and receiver.

RSU (%)	msg	LTE	WiFi	DSRC
sender	*BSM*	1	2.63	**1.036**
	image	120.53	**100.6**	-
receiver	*BSM*	1.18	120.53	**0.1**
	image	1	9.03	-

Table 8: The memory footprint of OBU as sender and receiver.

OBU (kb)	msg	LTE	WiFi	DSRC
sender	*BSM*	51832	51968	**20100**
	image	**165996**	312400	-
receiver	*BSM*	53408	51924	**10224**
	image	**126380**	126688	-

Table 9: The CPU utilization of OBU as sender and receiver.

OBU (%)	msg	LTE	WiFi	DSRC
sender	*BSM*	1.2	1.11	**1.08**
	image	**91.44**	99.76	-
receiver	*BSM*	1.25	1.1	**0.124**
	image	**1.36**	1.78	-

References

[1] Google's Waymo invests in LiDAR technology, cuts costs by 90 percent, 2017. Available at https://arstechnica.com/cars/2017/01/google-s-waymo-invests-in-lidar-technology-cuts-costs-by-90-percent/.

[2] Global Autonomous Driving Market Outlook, 2018, 2018. Available at https://www.prnewswire.com/news-releases/global-autonomous-driving-market-outlook-2018-300624588.html.

[3] Robot Operating System(ROS), Powering the World's Robots, 2019. Available at https://www.ros.org/.

[4] USRP B210, 2019. Available at https://www.ettus.com/all-products/ub210-kit/.

[5] Min Chen and Yixue Hao. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE Journal on Selected Areas in Communications*, 36(3):587–597, 2018.

[6] Nima Eshraghi and Ben Liang. Joint offloading decision and resource allocation with uncertain task computing requirement. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1414–1422. IEEE, 2019.

[7] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D Sutton, Pablo Serrano, Cristina Cano, and Doug J Leith. srsLTE: an open-source platform for LTE evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, pages 25–32, 2016.

[8] Jason M Hirst, Jonathan R Miller, Brent A Kaplan, and Derek D Reed. Watts Up? Pro AC power meter for automated energy recording, 2013.

[9] Shinpei Kato, Eijiro Takeuchi, Yoshio Ishiguro, Yoshiki Ninomiya, Kazuya Takeda, and Tsuyoshi Hamada. An open approach to autonomous vehicles. *IEEE Micro*, 35(6):60–68, 2015.

[10] John B Kenney. Dedicated Short-Range Communications (DSRC) standards in the United States. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[11] Liangkai Liu, Yongtao Yao, Ruijun Wang, Baofu Wu, and Weisong Shi. Equinox: A road-side edge computing experimental platform for CAVs. In *Proceedings of third IEEE International Conference on Connected and Autonomous Driving(MetroCAD)*, 2020.

[12] Liangkai Liu, Xingzhou Zhang, Qingyang Zhang, Andrew Weinert, Yifan Wang, and Weisong Shi. Auto-VAPS: an IoT-enabled public safety service on vehicles. In *Proceedings of the Fourth Workshop on International Science of Smart City Operations and Platforms Engineering*, pages 41–47, 2019.

[13] Shaoshan Liu, Liyun Li, Jie Tang, Shuang Wu, and Jean-Luc Gaudiot. Creating autonomous vehicle systems. *Synthesis Lectures on Computer Science*, 6(1):i–186, 2017.

[14] Shaoshan Liu, Liangkai Liu, Jie Tang, Bo Yu, Yifan Wang, and Weisong Shi. Edge computing for autonomous driving: opportunities and challenges. *Proceedings of the IEEE*, 107(8):1697–1716, 2019.

[15] Navid Nikaein, Mahesh K Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. OpenAirInterface: A flexible platform for 5G research. *ACM SIGCOMM Computer Communication Review*, 44(5):33–38, 2014.

[16] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[17] SAE Surface Vehicle Standard. Dedicated Short Range Communications (DSRC) message set dictionary. *SAE International, November*, 2009.

[18] Steve LeVine. What it really costs to turn a car into a self-driving vehicle, 2017. Available at https://qz.com/924212/what-it-really-costs-to-turn-a-car-into-a-self-driving-vehicle/.

[19] Tuyen X Tran, Kevin Chan, and Dario Pompili. COSTA: Cost-aware service caching and task offloading assignment in mobile-edge computing. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2019.

[20] Junjue Wang, Ziqiang Feng, Shilpa George, Roger Iyengar, Padmanabhan Pillai, and Mahadev Satyanarayanan. Towards scalable edge-native applications. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 152–165, 2019.

[21] Yifan Wang, Liangkai Liu, Xingzhou Zhang, and Weisong Shi. HydraOne: an indoor experimental research and education platform for CAVs. In *2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19)*, 2019.