An Automotive Cooperative Collision Avoidance Service Based on Mobile Edge Computing

A. García Olmos, F. Vázquez-Gallego, R. Sedar[(✉)], V. Samoladas, F. Mira,
and J. Alonso-Zarate

Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA),
Av. Carl Friedrich Gauss 7, 08860 Castelldefels, Barcelona, Spain
{agarcia,francisco.vazquez,roshan.sedar,vsamoladas,fermin.mira,
jesus.alonso}@cttc.es

Abstract. Even before 5G is rolled out, Mobile Edge Computing
(MEC) can be considered as a key driver towards the deployment of
vehicular use cases, which pose stringent latency and bandwidth require-
ments to the underlying Vehicle-to-Everything (V2X) communication
infrastructure. In this paper, we present a MEC-enabled cooperative
Collision AVoidance (CAV) service designed to anticipate the detection
and localization of road hazards by extending vehicles' perception range
beyond the capabilities of their own sensors. The CAV service is a soft-
ware application that runs on MEC servers allocated at the roadside
and at Mobile Network Operators' (MNO) infrastructures. The CAV
service receives ETSI ITS-G5 standard-compliant messages transmitted
by vehicles: periodic Cooperative Awareness Messages (CAM), which
include the position, velocity and direction of the vehicle; and event-
triggered Decentralized Environmental Notification Messages (DENM),
which include the position of detected road hazards. The CAV service
creates a distributed dynamic map using all the received information,
and sends unicast messages to each vehicle with the relevant informa-
tion within its collision risk area. We have implemented and validated
the operation of the CAV service using vehicles' On-Board Units (OBU)
based on OpenC2X, an open-source experimental platform supporting
the ETSI ITS-G5 standard.

Keywords: V2X · MEC · Collision Avoidance · OpenC2X

1 Introduction

In recent years, the road safety has received much attention in the wake of
alarming figures by the World Health Organization[1] showing that 1.4 million
people were killed by road accidents in 2016. In order to improve road safety,

---

[1] https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

vehicles are becoming more aware of their environment thanks to the integration of smart sensors (i.e., cameras, radar, lidar, ultrasonic sensors, etc.) and their level of automation is rapidly increasing. In addition, in near future the use of Vehicle-to-Everything (V2X) communication will allow vehicles to exchange information with other vehicles (V2V), roadside infrastructure (V2I), pedestrians (V2P) or communication networks (V2N). Therefore, V2X communication will facilitate vehicles to cooperate with each other and to extend their perception range beyond the capabilities of their sensors, supporting vehicular services that increase traffic efficiency and road safety, and it will pave the way for future Cooperative-Intelligent Transportation Systems (C-ITS) and fully-automated driving.

The deployment of vehicular services will impose very strict requirements to the underlying communication infrastructure, which must guarantee that information reaches its destination in a secure and timely manner. Mobile Edge Computing (MEC) can be considered as a key enabler for V2X communication [1]. This is due to the fact that MEC brings processing, storage and networking resources closer to the network edge and facilitates the compliance of low-latency and high-bandwidth requirements. Furthermore, MEC servers can be easily installed at the roadside infrastructure and integrated within the Radio Access Network, deployed between the core network and cellular base stations [2].

Motivated by this context, we have designed and implemented a MEC-based Collision AVoidance (CAV) service that allows to anticipate the detection and localization of road hazards. The CAV service is a software application that runs on the MEC infrastructure with the aim of facilitating smooth vehicle reactions in order to avoid hard braking and collisions. Vehicles send their status information and detected hazards to the CAV service, which stores all collected information and sends to each vehicle the relevant data within its collision risk area. In this work, we describe the functionalities, networking and software architecture of the CAV service, and we present the implementation and validation of a proof-of-concept. This work is an extension of the demo presented in [3].

The remainder of this paper is organized as follows. Section 2 provides an overview on V2X communication, open-source implementations of the ITS-G5 protocol stack and field trials of vehicular use cases implemented with the support of MEC. Section 3 describes the scenarios of the cooperative collision avoidance use case. Sections 4 and 5 describe the design and implementation of the CAV service, respectively. Finally, Sect. 6 concludes this paper.

2   Related Work

In the context of C-ITS, two standards have been developed for direct short-range communications (DSRC): the ITS-G5 standard developed in Europe [4] and the IEEE WAVE standard developed in North America [5]. ITS-G5 and WAVE are based on IEEE 802.11p, which facilitates the formation of wireless ad-hoc networks whenever vehicles or roadside units are within the range of each other, thus enabling vehicles to directly communicate with each other (V2V) and with the roadside infrastructure (V2I).

The realization of the ITS-G5 protocol stack can be found in open-source implementations such as Vanetza[2] and OpenC2X [6]. These two experimental and prototyping platforms implement the majority of modules of the ITS-G5 reference architecture. For instance, OpenC2X does not include the security and GeoNetworking modules, but implements all the other modules and includes an interface to access the vehicle's CAN bus using an OBD2 adapter, whereas Vanetza leaves out the implementation of the Local Dynamic Map (LDM) and the interface to the vehicle's CAN bus. Both OpenC2X and Vanetza support the Cooperative Awareness Messages (CAM) and Decentralized Environmental Notification Messages (DENM) defined in the ITS-G5 facilities layer. CAM messages are transmitted periodically, including the vehicle's position, speed and direction, and DENM messages are event-triggered to disseminate cooperative awareness information like the position of road hazards.

In the absence of short-range wireless connectivity based on IEEE 802.11p, vehicles could be opted to communicate over the cellular networking infrastructure. Recently, 3GPP has developed the first set of cellular standards for V2X communication. Today's realization of Cellular-V2X (C-V2X) is based on LTE-V2X (3GPP Release 14, March 2017) and it will evolve into future 5G-V2X (3GPP Release 16, to be completed by end of 2019), also known as 5G New Radio, to support both cellular V2N and V2V communication. C-V2X is gaining support from the leaders of automotive and telecom industries, which has led to worldwide trials [7] of C-V2X, whereas the trials of DSRC have been limited to Europe, North America, and Japan.

As ETSI MEC outlines [8], the amount of data is expected to increase rapidly from connected vehicles and, therefore, these data sets can be conveniently collected and processed closer to the user, i.e., at the edge of the network. Clearly, such an architecture is ideal for building real-time situational awareness and High Definition maps by effectively using real-time data collection and fusion from multiple available sources, e.g., vehicles' on-board sensors and road-side units. To this end, a number of MEC-based architectures have already been proposed in the existing literature. The works in [9,10] suggest that MEC-assisted network architectures will likely contribute to reduce end-to-end latency in V2X communication. The authors of [11] propose a MEC-based architecture for cellular V2N networks where MEC servers are integrated within the Radio Access Network and deployed between the core network and base stations.

Recently, several field trials have demonstrated some vehicular use cases with the support of MEC [7]. A real use case deployment of MEC-based architecture can be found in the Car2MEC [12] project, in which a MEC infrastructure has been deployed covering 30 Km on the A9 motorway in Germany to demonstrate time-critical use cases like emergency warning, end of jam warning and variable speed limit assistant, sending information from one car to another with a delay below 20 ms using LTE. Another field trial presented in [13] demonstrated the use case of assistance in road intersections.

---

[2] https://github.com/riebl/vanetza.

3   Cooperative Collision Avoidance Use Case

The Cooperative Collision Avoidance use case refers to the detection of road hazards that are beyond the perception range of vehicles' on-board sensors as well as keep vehicles informed on potential collision risk events ahead of time. Under certain conditions, typical sensors (e.g., radar, camera and Lidar) may not be able to detect some hazardous events with enough level of anticipation, which may lead to dangerous maneuvers or hard braking of vehicles and potentially cause a collision. As an example, typical scenarios of this use case are the situations where there is an incident (e.g., car accident, traffic jam, etc.) which is not clearly visible because of a downhill or behind a curve, or there is an obstacle like a building that blocks the view of approaching vehicles to an intersection.

In this Cooperative Collision Avoidance use case, vehicles cooperate with each other through the dissemination of their own status information (i.e., position, velocity and direction) as well as the location of the events detected by the sensors. This cooperation enabled by V2X communication that allows vehicles to anticipate the detection and localization of such dangerous events, and facilitates smooth vehicle reactions to avoid hard braking or to escape potentially from a collision.

4   Architecture of the Collision Avoidance Service

In this work, we consider cellular base-stations (BS) that provide wireless connectivity to the vehicles driving within their coverage area. Each cellular BS is connected to one MEC server where the CAV service is deployed. MEC servers are connected to a centralized public cloud Data Center (DC) through the core network, and the DC keeps MEC servers with up-to-date information. To this end, we implement a centralized MQTT[3] broker service from which the MEC servers communicate via the broker deployed in the Cloud Data Center.

Each vehicle is equipped with a cellular User Equipment (UE), a computer, a Human Machine Interface (HMI) and a set of sensors that allow detecting road hazards in the proximity, such as icy surfaces, unexpected manoeuvres, traffic jams, emergency braking, etc. The communication between the CAV service and vehicles is based on V2N with the use of ITS-G5 standard-compliant messages: periodic CAMs and event-triggered DENMs. Vehicles are able to receive up-to-date information through frequent communication with the MEC server where the CAV service is deployed.

Figure 1 illustrates the software modules of the CAV service. The communication between the modules is realized using the ZeroMQ[4] asynchronous messaging library. The CAV service performs the following operations: (1) it receives and decodes CAM and DENM messages transmitted by vehicles, (2) it stores vehicles' status and hazards related information in a database, (3) it calculates distances between vehicles, and between vehicles and hazards, and (4) it encodes

---

[3] http://mqtt.org/.
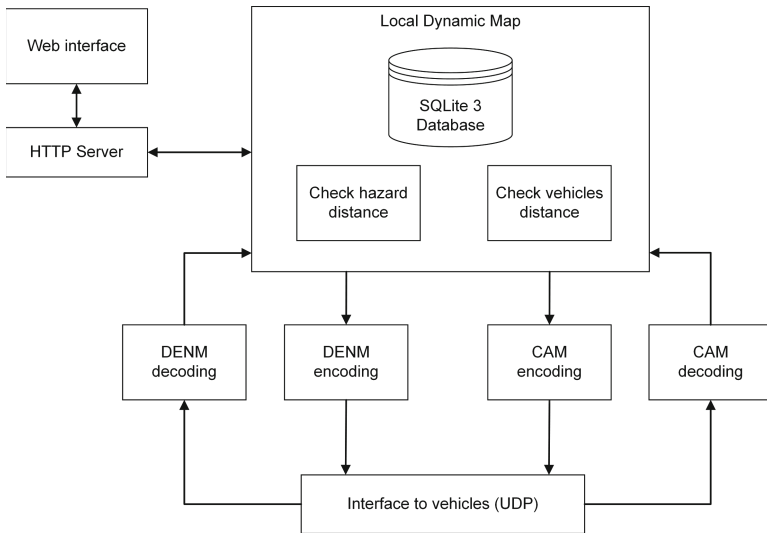
[4] http://zeromq.org/.

Fig. 1. Software architecture of the Collision Avoidance service

and transmits CAM and DENM messages only to those vehicles that are either approaching road hazards or other vehicles. As shown in Fig. 1, incoming messages are received at the CAV service through a UDP server included in the Interface-to-Vehicles module. At the UDP server, the incoming CAM and DENM messages are handled separately and forwarded into their respective CAM and DENM modules, which decode the messages and forward the extracted information to the LDM module. The LDM module stores the most recent information in a SQLite[5] database system to keep up-to-date events.

Two parallel threads are executed concurrently inside the LDM module. One thread computes the distance between each pair of vehicles, while the other thread computes the distance between each vehicle and each road hazard. If the distance between any two vehicles is smaller than a pre-defined threshold, e.g., 100 m, the CAV service will transmit a CAM message to both vehicles. If the distance between a vehicle and a hazard is below a certain threshold, e.g., 50 m, the CAV service will transmit a DENM message to the vehicle. The CAM and DENM messages are transmitted over UDP through the Interface-to-Vehicles module.

The CAV service integrates a web interface module (over HTTP) that allows to visualize the positions of vehicles and hazards over OpenStreetMap[6], configure the distance threshold of the collision risk area of each vehicle, and clear the information stored in the database.

---

[5] https://www.sqlite.org.
[6] https://www.openstreetmap.org/.

5   System Validation

We have implemented and validated a proof-of-concept of the CAV service in a small scale testbed using WiFi. The testbed is arranged using two Intel NUC[7] computers. The CAV service runs on Ubuntu 16.04 environment in one of the NUC computers, which operates as the MEC server and the WiFi access point. In the other NUC computer we have deployed three separate Virtual Machines (VMs) with Ubuntu 16.04. Each VM runs one vehicle application in it. In this way, we have deployed three virtualized vehicles and they communicate with the CAV service via WiFi.

We have developed a vehicle application based on OpenC2X which implements the ITS-G5 protocol stack, including a graphical user interface (GUI) with an LDM and allows to manually trigger simulated events. We have extended OpenC2X with a new software component that encapsulates the CAM and DENM messages over UDP in the transport layer. Since we use virtualized vehicles, instead of acquiring real positions from a GNSS receiver, the vehicle application reads the position of the vehicle from a trajectory file that contains a sequence of geographical coordinates with associated time-stamps. The vehicle application allows to select a trajectory file, start, pause and stop the movements of the vehicle. The vehicle application transmits CAM messages to the CAV service every 100 ms and sends a DENM when it is manually triggered.

We have demonstrated that if two vehicles approach at a distance below a predefined threshold, the CAV service transmits CAM messages to both vehicles, which show the position of the other vehicle on their LDM. Finally, we have generated simulated events and validated that if a vehicle approaches to any hazard at a distance below a certain threshold, the CAV service transmits a DENM message to that vehicle, which shows a warning on its GUI.

6   Conclusions

In this paper, we have presented a new MEC-based cooperative Collision Avoidance (CAV) service for vehicular networks. The CAV service is a software application deployed in the MEC infrastructure. Using Vehicle-to-Network (V2N) communication, vehicles exchange information with the CAV service in the form of unicast CAM and DENM messages over UDP. The incoming messages from vehicles are processed by the CAV service, which disseminates to each vehicle only the relevant information within its collision risk area (e.g., dangerous events, presence of other vehicles, etc.). We have implemented a proof-of-concept of the CAV service and demonstrated its functionalities under different traffic conditions in a small scale testbed. As a future work, a field trial of the CAV service will be deployed in the city of Barcelona using LTE Small Cells at 3.5 GHz in order to evaluate key performance metrics under different real road traffic conditions. We plan to evaluate the performance of the CAV service in terms of latency, scalability and robustness against failures in the MEC servers.

---

[7] https://www.intel.com/content/www/us/en/products/boards-kits/nuc.html.

References

1. Shah, S.A.A., et al.: 5G for vehicular communications. IEEE Commun. Mag. **56**, 111–117 (2018)
2. Li, L., et al.: A novel mobile edge computing-based architecture for future cellular vehicular networks. In: IEEE WCNC, pp. 1–6, March 2017
3. Vázquez-Gallego, F., et al.: Demo: a MEC-based collision avoidance system for future vehicular networks. In: CNERT INFOCOM Workshop, April 2019
4. Wevers, K., et al.: V2X communication for ITS - from IEEE 802.11p towards 5G, June 2017. https://futurenetworks.ieee.org/tech-focus/march-2017/v2x-communication-for-its
5. Task Group p: IEEE P802.11p: Wireless Access in Vehicular Environments (WAVE). draft standard ed. (2006)
6. Klingler, F., et al.: Field testing vehicular networks using OpenC2X. In: 15th ACM International Conference on Mobile Systems, pp. 178–178, June 2017
7. 5GAA: Toward fully connected vehicles: edge computing for advanced automotive communications. White Paper (2017)
8. Reznik, A., et al.: Developing software for multi-access edge computing. ETSI White Paper, September 2017
9. Emara, M., et al.: MEC-assisted end-to-end latency evaluations for C-V2X communications, pp. 1–9. IEEE (2018)
10. Datta, S.K., et al.: Vehicles as connected resources: opportunities and challenges for the future. IEEE Veh. Technol. Mag. **12**, 26–35 (2017)
11. Li, L., Li, Y., Hou, R.: A novel mobile edge computing-based architecture for future cellular vehicular networks. In: 2017 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6, March 2017
12. Car2MEC Project, September 2018. https://www.nokia.com/about-us/news/
13. Press release: Comba Telecom and ASTRI demonstrate Mobile Edge Computing system (MEC) with Vehicle-to-Everything (V2X) system prototype at Mobile World Congress 2017 in Barcelona