OScar: An ETSI-Compliant C-ITS Stack for Field-Testing with Embedded Hardware Devices

Marco Rapelli
Politecnico di Torino
Turin, Italy
marco.rapelli@polito.it

Francesco Raviglione
Politecnico di Torino
Turin, Italy
francesco.raviglione@polito.it

Claudio Casetti
Politecnico di Torino
Turin, Italy
casetti@polito.it

***Abstract*—Vehicular communication technologies (also known as Vehicle-to-Everything, V2X) have emerged as a solution to enable not only higher levels of automation, but also to improve the safety and environmental impact of road transportation systems. To this aim, the European Telecommunication Standards Institute (ETSI) has defined a set of standards detailing several message types for vehicular communications, along with a full networking stack to enable vehicles to communicate between each other and with the infrastructure. However, there is currently a critical scarcity of open stack implementations for research and testing of innovative V2X solutions in the field. To tackle this issue, this paper presents OScar, an open-source, ETSI-compliant C-ITS stack for embedded hardware devices aimed at vehicular field tests. Addressing the lack of affordable, customizable software for V2X communications, OScar proposes a lightweight, self-contained, Linux-based solution, designed to run even in low-end hardware. OScar has been validated in the field thanks to a series of preliminary tests involving Cooperative Awareness Messages disseminated through IEEE 802.11p.**

***Index Terms*—Vehicle-to-Everything communication, ITS stack implementation, Vehicular field tests, ETSI C-ITS**

I. Introduction and Motivation

With the increase of population in urban areas, traffic congestion issues and road accidents have drastically increased. As stated by the "Statistics of Road Traffic Accidents" report [1] from the United Nations Economic Commission for Europe, the number of fatalities occurred in road accidents in Europe and North America is greater than 80 thousand units for 2021. This is the equivalent of 225 casualties per day. Improving safety on the roads for both vehicles and pedestrians has thus become a crucial goal in recent years.

In an attempt to mitigate these serious problems, the European Telecommunication Standardization Institute (ETSI) has increased the effort in the standardization of vehicular messages and applications. Since the first standards for a broad Intelligent Transportation System (ITS) and the corresponding ETSI Facilities Layer, many simulation studies have proven that vehicular applications can actually reduce the impact of traffic and pollution in urban areas [2], [3] as well as mitigate the number and the impact of accidents on the road [4]–[6].

Simulations represent a very valid solution for the initial experimentation on vehicular applications, especially on a large scale. However, it is well known that tests on the field using car equipped with vehicular communication capabilities produce results with a much greater impact, since they are affected by issues that simulation models may not be able to fully capture. These include real-world environmental factors, unpredictable human behavior, and the complex dynamics of live traffic conditions. However, building a setup for a field test using cars equipped with hardware boards dedicated to vehicular communication is not trivial and the few commercial devices available for this purpose are quite expensive and with little customization options. To tackle the lack of affordable and customizable hardware for V2X, a few works leveraged generic embedded hardware boards, which may not be specifically designed for working in vehicular environments, but that can be customized to let them fully support the frequencies defined by the ETSI standards. Relevant examples are the OpenWrt-V2X project [7] and the work by Richter *et al.* [8]. Nevertheless, there is still an issue: the literature lacks lightweight, self-contained, open Cooperative ITS (C-ITS) full stack implementations for embedded boards, which can be used in both commercial and open Linux-based solutions for on-field V2X communications.

To address this problem, this short paper presents OScar (Open Stack for car), an open-source, ETSI-compliant C-ITS stack implementation, designed for being exploited in embedded hardware devices for vehicular field tests. OScar is a currently ongoing project of a full implementation of the ETSI C-ITS stack, for the management, transmission and reception of vehicular messages. At the time of this writing, OScar supports the full Basic Services for Cooperative Awareness Messages (CAMs) and Vulnerable Road Users Awareness Messages (VAMs), while in the near future we envision to provide support for Decentralized Environmental Notification Messages (DENMs), Infrastructure to Vehicle Information Messages (IVIMs), Cooperative Perception Messages (CPMs) and for security header and certificate formats. In addition, we also plan to support Electric Vehicle Charging Spot Notification (EVCSN) messages. All the ETSI standards considered for the implementation of the mentioned messages

TABLE I
ETSI standards of considered messages in OScar.

ETSI messages	ETSI standard
Cooperative Awareness Message (CAM)	ETSI EN 302 637-2 V1.4.1
Vulnerable Road User Awareness Message (VAM)	ETSI TS 103 300-3 V2.1.1
Decentralized Environmental Notification Message (DENM)	ETSI EN 302 637-3 V1.3.1
Infrastructure to Vehicle Information Messages (IVIMs)	ETSI TS 103 301 V1.3.1
Collective Perception Message (CPM)	ETSI TR 103 562 V2.1.1
Security headers and certificate formats	ETSI TS 103 097 V2.1.1
Electric Vehicle Charging Spot Notification (EVCSN)	ETSI TS 101 556-1 V1.1.1

are reported in Table I.

OScar extends the existing literature as follows: *(i)* it provides an ongoing implementation of an ETSI-compliant C-ITS stack suited for Linux-based embedded hardware boards for vehicular field tests; *(ii)* it has the support for the most safety-relevant ETSI Basic Services, with an ongoing activity to support the whole ETSI C-ITS stack. This differs from existing projects in the literature providing only the message encoding/decoding part of the Basic Services; *(iii)* it is a completely open-source project under a GPLv2 license[1]; *(iv)* to the best of our knowledge it is the first open framework for vehicular field tests *with full Basic Services* to be directly derived from a vehicular simulator (i.e., ms-van3t [9]), letting the user easily port the code developed in the simulated environment to embedded hardware devices running OScar; *(v)* it is a self-contained, lightweight, single-executable tool that can be easily cross-compiled and integrated in embedded Linux distributions such as OpenWrt.

The rest of the paper is organized as follows: Section II discusses other examples of commercial and open solutions for ETSI C-ITS stack implementations. Section III describes the framework of OScar, while Section IV shows some preliminary results of CAMs dissemination. Eventually, Section V concludes and summarize the work.

II. Related Work

Nowadays, a number of commercial solutions exist providing an ETSI C-ITS stack for the On-Board Units (OBUs) of commercial vehicles. Producers of OBUs include Cohda Wireless, Commsignia, Chemtronics, and Danlaw. Usually, commercial units come with efficient, yet expensive SDKs (Software Development Kits), enabling the reception and transmission of different types of ITS messages. However, they are usually proprietary software, providing little customization outside the standards, thus not enabling the research of innovative, or not yet standardized, approaches. For instance, while it is possible to configure whether secured or non-secured messages should be transmitted, it is much more complex to test new protocols or to change the frequency of CAMs to test the effect of higher update rates. The latter was of particular interest for a number of corridor projects, such as the 5G-CARMEN European project, aiming at testing cross-border 5G-enabled services with an update rate (thus, a CAM frequency) of 20 Hz [10].

As opposed to commercial solutions, OScar proposes a fully open stack, based on the codebase of the ms-van3t V2X simulator and emulator [9], with vast room for customization, and specifically tailored at testing innovative approaches and protocols in the field.

Indeed, besides the commercial SDKs, there is currently a scarcity of open-source ETSI C-ITS solutions in literature. The most notable example is probably Vanetza [11], providing a full ITS stack implementation complete with the BTP and GeoNetworking layers. An extension of Vanetza, called NAP-Vanetza, has also been developed by Instituto de Telecomunicações in Portugal, providing additional features on top of Vanetza and a more extensive set of supported ETSI messages [12]. These include Vulnerable road users Awareness Messages (VAMs) that are gaining more and more interest as they enable several critical road safety use cases, Maneuver Coordination Messages (MCMs) that were recently defined by ETSI for managing complex maneuvers among two or more vehicles on the road (i.e., lane merging, overtaking, etc.), and many others.

However, both Vanetza and NAP-Vanetza, while they support the encoding and decoding of a large set of messages, they still miss full-fledged Basic Services implementations, which define the rules for the generation of the messages to be sent. OScar, instead, has the full service implementation for the Cooperative Awareness (CA) Basic Service and the Vulnerable Road User (VRU) Basic Service, envisioning to extend this to other important ETSI services such as the Decentralized Environmental Notification (DEN) Basic Service and the Collective Perception (CP) Basic Service.

It is also important to mention that the OScar project stems from the ms-van3t implementation and, for this reason, they have a common codebase structure. Thanks to that, and to how the models in ms-van3t are similar to the Linux kernel implementations, the porting of a piece of code from the ms-van3t simulator to the OScar framework for field tests can be done easily with few steps. This makes the two projects a significant example of paired software for vehicular application both on simulated and real-world environment.

Finally, it is worth highlighting an open implementation of BTP and GeoNetworking, partly developed as part of the i-GAME European project [13]. Both the networking and transport layers have been developed in Java, in addition to a partial implementation of the Facilities Layer, currently supporting CAMs and DENMs. Unlike this project, OScar is developed using the latest C++ revision, to enable an easier, more efficient integration into low-power embedded devices. It is also designed to provide an all-in-one solution including the creation and management of sockets for the transmission of packet over any cabled or wireless interface. Finally, it provides out-of-the-box support for patched Linux kernels enabling communication via IEEE 802.11p (i.e., the direct "vehicular WiFi"), such as the one included in the OpenWrt-V2X 21.02 project [7], [14].

III. Framework description

The idea behind the development of OScar comes from the lack of open-source, fully customizable ETSI C-ITS stack. Customizability is indeed a crucial feature for the experimentation of novel protocols and services for V2X. However, it is often very limited in existing commercial solutions. Additionally, there is a necessity for lightweight tools that can be easily cross-compiled and integrated even in low-end, low-power hardware devices. Motivated by this issue, and starting from the OpenWrt-V2X project [7], [15], OScar was initially developed as a C++ framework including a full CA Basic Service, and a Local Dynamic Map (LDM) implementation (based on the ETSI LDM standard definition [16]) with a web-based user interface, based on the Server Local Dynamic Map first presented in [10]. OScar is also based on the Open CA Basic Service (OCABS) project [17] and the Automotive Integrated Map (AIM) service [14]. It was later extended to include other Basic Services and features, including the support to VRUs.

OScar includes the ITS Transport and Networking layer, comprising the Basic Transport Protocol (BTP) and GeoNetworking (currently supporting GeoBroadcast and Single Hop Broadcast). On top of the Transport and Networking layer,

[1]https://github.com/DriveX-devs/OScar

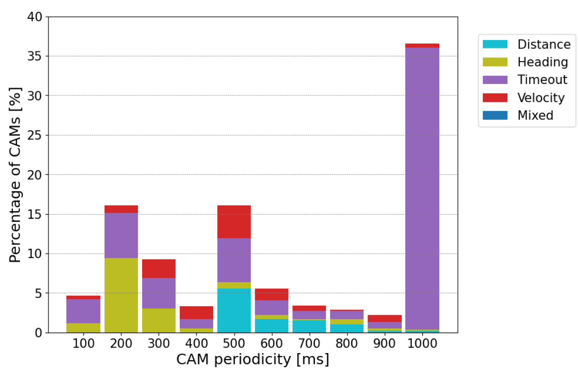
(a) (b) (c)

Fig. 1. Vehicular field tests setup. (a) APU2C4 embedded board. (b) From bottom left to upper right: the ArduSimple kit, the APU2C4 acting as our OBU and the external power supply. (c) From left to right: the magnetic GNSS antenna, the legacy radio antenna of the car and the two magnetic DSRC antennas.

OScar implements the full Facilities Layer, leveraging a modular, easily customizable code structure, including an efficient LDM, than can be accessed by other services through a JSON-over-TCP interface. Concerning instead the lower layers, OScar uses raw Linux sockets to interface with any wireless or wired interface available in the Operating System (OS). These may include IEEE 802.11p interfaces, as in [15]. Support to 3GPP Release 14 LTE-V2X is also currently being developed, thanks to the integration of functions from the Qualcomm SDK for C-V2X modules.

One of the major features of OScar is how it provides a self-contained project, with very little external dependencies. This simplifies the cross-compilation procedure for different kinds of architectures. Indeed, at the time of this writing, the only dependency required outside the OScar source code is *libgps*, that is normally available on any Linux distribution supporting the *gpsd* service. The generated binary file, being as light as just 2.5 MB, can then be easily deployed to multiple devices at once. Finally, to simplify the collection of data in the field, OScar includes advanced logging features, for instance concerning the CAM and VAM generation.

(a)



(b)

Fig. 2. Vehicular dissemination results in an urban field test. (a) CAMs traces sent. (b) Percentages of CAMs sent with different frequencies.

IV. Results

To initially validate the OScar implementation and to show the advantages it may bring to V2X research, we have performed a series of preliminary field tests focused on CAMs dissemination. The field tests have been performed with a dedicated setup, shown in Fig. 1. In particular, Fig. 1a depicts an APU2C4 embedded board running the latest version of OpenWrt-V2X (21.02.1), a special adaptation of the OpenWrt Linux OS supporting communication via IEEE 802.11p at 5.8/5.9 GHz [15], with supported modules, such as the selected Mikrotik R11E-5HND. As displayed in Fig. 1b, we coupled our board with an external power supply and with an ArduSimple simpleRTK2B Fusion board with 4G NTRIP master providing GNSS positioning thanks to Real Time Kinematic (RTK) service and ensuring centimeter-level accuracy by receiving the position corrections from a base station via LTE. Eventually, we mounted a GNSS magnetic antenna and two DSRC antennas on the roof of our vehicle, as can be seen in Fig. 1c.

Our field test setup enabled us to retrofit a legacy car with vehicular communication systems and drive through Turin, Italy, broadcasting ETSI CAM messages thanks to the OScar CA Basic Service. Fig. 2 shows the results of the CAM dissemination while driving our equipped vehicle around a block. Every colored dot depicted in Fig. 2a represents the position of a CAM sent and their colors match the ones in Fig. 2b, where the percentage of CAMs sent with different frequencies and for different motivations is inspected.

As can be seen, the obtained results validate the capability of OScar to properly disseminate CAMs following the ETSI standards. CAMs are disseminated with a variable periodicity between 100 ms and 1 s. This is due to ETSI standards foreseeing the immediate transmission of CAMs every time the speed, heading and position difference with respect to the last CAM exceeds the thresholds, respectively, of 0.5 m/s, 4 degrees and 4 meters, checked every 100 ms. This can be clearly seen in Fig. 2a. When the car is turning, CAMs are mostly sent due to significant heading variations (yellow dots), while the most common triggering condition in straight segments is the position difference (light blue dots), due to the vehicle travelling at a higher speed. Finally, it is possible to notice that more CAMs were sent while braking and accelerating (red dots), for instance in the roundabout, consistently with the standard.

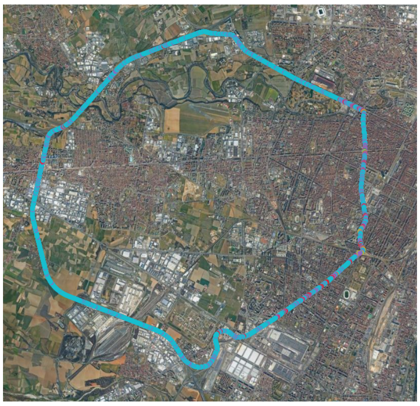
In the same spirit, we also performed a more extensive field test. Fig. 3 shows the results of a CAM dissemination while driving in a mixed urban and highway scenario along the Turin ring road (as depicted in Fig. 3a). Comparing the percentage of CAMs sent with different frequencies in Fig. 2b and Fig. 3b, it is possible to notice how, in the urban scenario, the majority of CAMs are sent at 1 Hz and with time motivation, due to the lower speeds and limited heading variations. On the other hand, in the mixed scenario, the transmission frequencies of CAMs are higher, as a consequence of the greater speed during the field test.

V. Conclusions

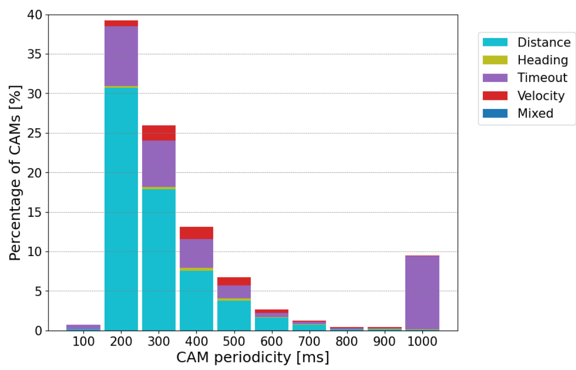
This paper has presented OScar, a fully open, self-contained and lightweight ETSI C-ITS stack implementation. OScar has been designed to tackle the critical lack of customizable software for V2X research and experimentation, while providing at the same time an all-in-one solution that can be easily integrated into Linux-based embedded devices.

We have validated OScar by means of a initial set of tests focusing on CAM dissemination, showing its compliance to the standard and how it can be used to retrofit unequipped vehicles for V2X research, when paired with proper embedded hardware.

OScar development is ongoing and the implementation is expected to support both a larger set of ETSI Basic Services as well as Cellular-V2X technology in the near future.



(a)



(b)

Fig. 3. Vehicular dissemination results in a mixed urban and highway field test. (a) CAMs traces sent. (b) Percentages of CAMs sent with different frequencies.

Acknowledgment

This work was partially funded by the European Union through the project CONNECT under grant agreement no. 101069688 and was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

References

[1] Economic Commission for Europe, *Statistics of road traffic accidents in Europe and North America*. UN, 2021, vol. 56.

[2] A. Jame, M. Rapelli, and C. Casetti, "Reducing pollutant emissions through Virtual Traffic Lights," *Computer Communications*, vol. 188, pp. 167–177, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422000937

[3] A. Bazzi, A. Zanella, and B. M. Masini, "A distributed virtual traffic light algorithm exploiting short range v2v communications," *Ad Hoc Networks*, vol. 49, pp. 42–57, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870516301548

[4] G. Avino, M. Malinverno, C. Casetti, C. F. Chiasserini, F. Malandrino, M. Rapelli, and G. Zennaro, "Support of safety services through vehicular communications: The intersection collision avoidance use case," in *2018 International Conference of Electrical and Electronic Technologies for Automotive*, 2018, pp. 1–6.

[5] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino, and S. Scarpina, "Edge-based collision avoidance for vehicles and vulnerable users: An architecture based on mec," *IEEE Vehicular Technology Magazine*, vol. 15, no. 1, pp. 27–35, 2020.

[6] I. Soto, M. Calderon, O. Amador, and M. Urueña, "A survey on road safety and traffic efficiency vehicular applications based on c-v2x technologies," *Vehicular Communications*, vol. 33, p. 100428, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209621000978

[7] (2023) GitHub - OpenWrt-V2X. [Online]. Available: https://github.com/francescoraves483/OpenWrt-V2X

[8] D. Richter, L. Pirl, J. Beilharz, C. Werling, and A. Polze, "Performance of Real-Time Wireless Communication for Railway Environments with IEEE 802.11p," in *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS)*, 2019. [Online]. Available: http://scholarspace.manoa.hawaii.edu/handle/10125/60190

[9] F. Raviglione, C. R. Carletti, M. Malinverno, C. Casetti, and C. Chiasserini, "ms-van3t: An integrated multi-stack framework for virtual validation of v2x communication and services," *Computer Communications*, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366424000227

[10] F. Raviglione, C. M. R. Carletti, C. Casetti, F. Stoffella, G. M. Yilma, and F. Visintainer, "S-ldm: Server local dynamic map for vehicular enhanced collective perception," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–5.

[11] R. Riebl, C. Obermaier, S. Neumeier, and C. Facchi, "Vanetza: Boosting research on inter-vehicle communication," *Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)*, pp. 37–40, 2017.

[12] (2024) GitLab - NAP-Vanetza. [Online]. Available: https://code.nap.av.it.pt/mobility-networks/vanetza

[13] A. Voronov. (2020) GitHub - ETSI ITS G5 GeoNetworking Stack. [Online]. Available: https://github.com/alexvoronov/geonetworking

[14] F. Raviglione, C. Casetti, and F. Restuccia, "Edge-v : Enabling vehicular edge intelligence in unlicensed spectrum bands," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–5.

[15] F. Raviglione, M. Malinverno, and C. Casetti, "Demo: Open source platform for ieee 802.11p nics evaluation," in *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2019, pp. 1–3.

[16] ETSI, "ETSI EN 302 895 V1.1.1 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)," European Telecommunication Standard Institute, European Standard, 2014.

[17] (2021) GitHub - OCABS-project. [Online]. Available: https://github.com/francescoraves483/OCABS-project