Demo: Open source testbed for vehicular communication

F. Raviglione, M. Malinverno, C. Casetti

***Abstract*—The challenge of enabling the communications between the vehicle and its surroundings is being faced by the entire automotive industry, while the main standardization bodies are undergoing a huge effort to propose new solutions and improve the existing ones. The lack of open source solutions for vehicular communications penalizes the technology advances, and for this reason we present an open source platform based on *PC Engines*' boards and *Unex*'s WNICs for the testing of V2X (vehicle-to-everything) applications. Our platform enables the connectivity over a 802.11p channel between two boards that can be deployed as wireless dongles, so it can be used to extend the network capabilities of any kind of computing system. The testbed has been setup to work with several applications: from video streaming, to online gaming, to a containerized version of a latency tester, called LaTe.**

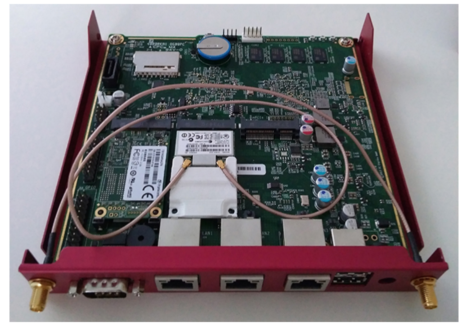

Fig. 1. APU1D board, where two additional components are visible: the Trascend 16GB MLC SSD, and the Unex DHXA-222 WNIC.

I. Introduction

The world of vehicular communications is experiencing a period of great interest from the academia and industry. 3GPP is developing its own cellular-based solutions, while IEEE is considering the introduction of a new vehicular amendment, called IEEE 802.11bd [1]. However, before any other under-development protocol hits the market, the choice for DSRC (Dedicated Short-Range Communications) will be legacy IEEE 802.11p. Despite being standardized almost ten years ago, it is not easy to find open source solutions for the assessment of vehicular applications. That is what drove us to the creation of an up-to-date, Linux based, open source platform that can be used with any IEEE 802.11p-compatible WNIC and that provides all the hardware and software elements needed to deploy vehicular testbeds like the one presented in this work. One of the features tested over our platform is the services delivery using Linux containers. In particular, we created a containerized version of an open source application based on the Latency Measurement Protocol (LaMP), called Latency Tester (LaTe) [2], that can be used to evaluate the application-layer latency between nodes, with the capability of testing the RTT and unidirectional latency using different EDCA's traffic classes.

II. Testbed description

The embedded boards that we used to develop the testbed are *PC Engines APU1D*, as shown in Figure 1, which support mPCIe cards, like *Unex's DHXA-222*, which are the WNICs of choice. These boards mount a AMD G-series dual-core T40E x86 CPU with 64 bits support and 2 GB of DRAM. They also offer the possibility to install an mSATA SSD as storage, increasing the memory performance, thus we opted for a SATA III Transcend MSA370 MCL NAND Flash SSD for each board. The Unex's WNICs that we used are based on the *Atheros AR9462* chipset, supported by the *ath9k* Linux driver. The OS deployed inside the motherboards is the latest version of OpenWrt, i.e., release 18.06.1, with Linux kernel 4.14.63. Our setup, which is available on GitHub under the GPLv2 license [3], is briefly described in the next subsections.

The *DHXA-222* WNICs manage the physical layer of our testbed. Their main characteristics are a dual-band support (over 2.4 and 5 GHz, including 5.8/5.9 GHz), MIMO 2x2 operations, 17 dBm maximum transmission power and their half-size mPCIe form factor. The usage of channels in the DSRC frequency band (5.8/5.9 GHz) has been enabled by patching the *ath9k* driver. This modification allows the user to select the ITS frequencies and to enable the Outside Context of a BSS (OCB) mode, along with the 10 MHz-wide channels, as foreseen by the standard. These patches were released in the OpenC2X embedded platform project, developed by *CCS Lab* in the University of Paderborn [4]. The work by Laux *et. al*, was ported, with minor modifications, to the latest stable version of OpenWrt, running on Linux kernel version 4.

Our work on MAC layer focused on the EDCA functionality of 802.11p protocol. For this purpose, the Linux *mac80211* wireless subsystem was patched, in order to enable the selection of any of the four EDCA Traffic Classes by working with a dedicated socket options (SO_PRIORITY). This patch was again ported from the OpenC2X project to OpenWrt 18.06.1.

The testbed has been validated through extensive trials, mostly involving UPD over IPv4. The tests were performed by directly deploying the applications inside the *PC Engines'* board, as well as using a software bridge to connect the boards' Ethernet and wireless interfaces to use them as 802.11p dongles.

III. Demo description

Our demo testbed will be deployed using: (i) Two *PC Engine's APU1D* equipped as described in Section II. (ii) Two Laptops connected through Ethernet to the *APU1D* boards, from which it will be possible to interact with the wireless subsystem. (iii) A third PC which will be connected with a spectrum analyzer, showing a proof of the spectrum usage. We plan to showcase three different demos:

- The first will let the attendees measure the latency between the two PCs, using a containerized version of the LaTe tool [2]. The containers are realized using Docker [5], and it will be possible to perform latency measurements using different options, logging the data to *csv* files, which will be parsed in real time by MATLAB.
- The second demo will involve a live feed of a video streaming application, in which the attendees can see a real deployment of a multimedia data flow through the network, using two web-cams.
- In the third demo, it will be possible to let the users experience a local multiplayer game, and show how 802.11p can be used to provide entertainment services.

IV. Demo requirements

We kindly ask the organizer to provide us with:

1) Power supply.
2) Three monitors.
3) Two regular office desk for the demo setup.

The demo setup will require up to 30 minutes.

V. Acknowledgments

This work has been performed in the framework of the EU Horizon 2020 project 5G-CARMEN co-funded by the EU under grant agreement No. 825012. The views expressed are those of the authors and do not necessarily represent the project. The Commission is not liable for any use that may be made of any of the information contained therein.

References

[1] IEEE. IEEE 802 pars & icaids under consideration [online]. http://www.ieee802.org/PARs.shtml
[2] LaTe + LaMP home page [online]. https://francescoraves483.github.io/LaMP_LaTe/
[3] Github - francescoraves483/openwrt-v2x [online]. https://github.com/francescoraves483/OpenWrt-V2X
[4] S. Laux, G. S. Pannu, S. Schneider, J. Tiemann, F. Klingler, C. Sommer, and F. Dressler, "Demo: OpenC2X - an open source experimental and prototyping platform supporting ETSI ITS-G5," in *2016 IEEE Vehicular Networking Conference (VNC)*, Dec 2016, pp. 1–2.
[5] Docker web page [online]. https://www.docker.com/