Autonomous Driving through Sensing and V2X

Joaquim Ramos

***Abstract*—The abstract goes here.**

***Index Terms*—V2X**

I. Introduction

Autonomous vehicles, sensing and V2X to improve the overall process

Autonomous vehicles are revolutionizing transportation as we know it. Self-driving vehicles hold the potential to make travel safer, more efficient, and accessible to all. Their success relies on their ability to perceive and understand their environment, much like a human driver.

Sensing technology acts as the core foundation for the operation of autonomous vehicles. These advanced cars rely on a network of sensors, including Light Detection And Ranging (LiDAR), radar, cameras, ultrasonic sensors, and precise Global Positioning System (GPS) systems. These sensors work together to act as the "eyes and ears" of the vehicle, constantly capturing and processing real-time information about the surrounding environment. Through this array of sensors, autonomous vehicles can understand various crucial details, like the movements of nearby vehicles and pedestrians, recognize traffic signals, and spot potential obstacles. This data-rich sensory input empowers autonomous vehicles to make well-informed decisions, navigate complex situations, and prioritize the safety of both passengers and other road users.

However, despite the promise of sensing technology, it comes with its own set of limitations. Factors like adverse weather conditions, low-light situations, and distinguishing between objects of different sizes and materials can pose significant obstacles to sensor technology. This is where Vehicle to Everything (V2X) communication steps in as a crucial complement, bridging the gaps left by these sensor limitations. V2X technology enables vehicles to interact with their environment, including communication with other vehicles and infrastructure elements. This type of communication facilitates the exchange of critical information, such as traffic conditions, road hazards, and upcoming signals and signs. Additionally, it promotes coordination between autonomous and human-driven vehicles, reducing conflicts and optimizing traffic flow.

In this work ....

II. Related work

Sensing, V2X applied to autonomous vehicles – anything about autonomous vehicles themselves?

The research outlined in [1] introduces AutowareV2X, a V2X communication module designed to be integrated with Autoware. While the paper provides valuable insights, it offers limited details regarding the system's implementation. It is mentioned that Autoware and AutowareV2X maintain loose coupling, utilizing a set of functions to establish an interface between the V2X communication stack (Vanetza) and the Robot Operating System (ROS)2-based Autoware. Additionally, the authors propose a dual-channel mechanism to ensure wireless link redundancy. This mechanism allows for the transmission of Collective Perception Messages (CPMs) through two distinct wireless technologies, Wi-Fi and 4G. The experiment results indicate that CPMs can be transmitted within approximately 30 milliseconds, demonstrating their utility in enabling collision avoidance maneuvers. However, it's worth noting that the paper lacks in-depth information regarding the experimental conditions and the methodology used to calculate latency, leaving room for further clarification.

The work in [2] proposes a V2X communication framework for the CARLA platform. This framework facilitates the exchange of sensory data through V2X communications, enabling the creation of simulated datasets for a wide range of scenarios.

III. Architecture of the proposed setup

A. Autoware and vehicle

B. Sensing

C. V2X

Regarding the V2X communication module, the vehicle is equipped with an On-board Unit (OBU) that plays a central role in facilitating V2X communication. The OBU integrates WAVE communication, enabling communication with Roadside Units (RSUs) and other vehicles. Within the OBU, Figure 1, four main services work in harmony to enable robust V2X capabilities. First, Vanetza takes charge of transmitting, receiving, encoding, and decoding vehicular network messages. The Message Queuing Telemetry Transport (MQTT) vehicle broker adopts the publisher/subscriber messaging pattern, allowing multiple services to subscribe to specific topics of interest and access real-time data efficiently. To ensure the persistence of real-time data, an SQLite database is integrated into the OBU. This feature is crucial as some services and applications may require access to a complete history of relevant data points. The database effectively stores this data for future reference and analysis. Finally, the message handler serves as the bridge between the vehicle and the generation of vehicular messages. It efficiently collects data from the vehicle, ensuring that the generated vehicular messages contain accurate and up-to-date vehicle information.

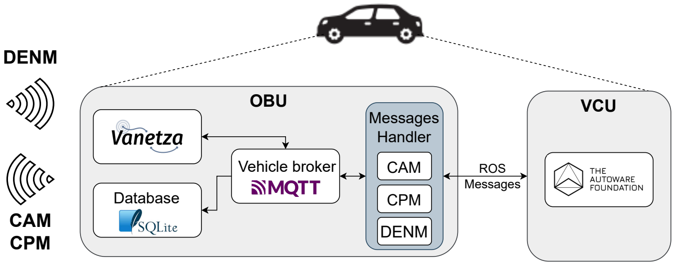

Fig. 1. OBU messaging pipeline architecture

D. Adapters

For each vehicular message, a dedicated adapter is created. These adapters have a crucial role: collect the necessary data and compile it into a structured JavaScript Object Notation (JSON) format containing all the essential message information. Once this compilation is complete, the data is then published on a specific MQTT topic that Vanetza subscribes to. Upon receiving a message, Vanetza efficiently encodes it and broadcasts it to all nearby nodes, ensuring widespread dissemination of the message.

1) CAM: Cooperative Awareness Message is a standardized message format defined by the European Telecommunications Standards Institute (ETSI) as part of the Intelligent Transport System (ITS) standards. CAMss play a crucial role in Cooperative Intelligent Transport Systems (C-ITS), facilitating communication between vehicles and infrastructure within vehicular networks. CAMss transmit essential information about a vehicle's current status, including:

- Basic Vehicle Information: This includes details such as the vehicle's identification (identifier) and type, such as whether it is a car, truck, motorcycle, or other road users like bicycles or pedestrians.
- Position and Orientation: CAMs provide accurate geographic position data using latitude and longitude coordinates. They also include information about the vehicle's heading or orientation, indicating its current direction of travel.
- Speed and Acceleration: CAMs convey information about the vehicle's current speed and acceleration, providing insights into its motion characteristics.
- Vehicle Dynamics: CAMs can include parameters related to vehicle dynamics, such as the yaw rate and wheel speeds. These details offer additional information about the vehicle's maneuvering and motion patterns.

CAMs are periodically transmitted by vehicles and RSUs within a single hop distance. The frequency of CAM transmission typically ranges between 1 Hz and 10 Hz. To construct a CAM, relevant vehicle status information must be collected. This information can be retrieved from the data available in Autoware. Therefore, the function of this particular adapter is to extract the pertinent information required for constructing CAM from Autoware and assemble the CAMs accordingly. Using the Autoware Simulator with a single moving vehicle, the information exchanged between different Autoware components was explored. After analyzing the message exchanges, it was determined that the topic $/awapi/vehicle/get/status$ contained the required information for CAMs. During the subsequent phase, a new component was developed with the primary objective of constructing CAM using the data obtained from Autoware. This module was designed to subscribe to a ROS topic, enabling it to gather the necessary information. Additionally, a new class was created, encompassing all the fields required for CAM generation. An object of this class was instantiated to store the information received from the ROS topic. Simultaneously, at a frequency of 10Hz, this adapter initiated the CAM creation process. The object containing the CAM information is converted into a JSON representation, which was subsequently published to a local MQTT broker located within the OBU. It is important to note that the NAP-Vanetza application also utilizes this local broker. Once NAP-Vanetza receives a CAM in the specified JSON format, compliant with the ETSI C-ITS standard, it proceeds with the encode and broadcast of the CAM to other Intelligent Transport System Station (ITS-S) instances within the network. Upon completion of the development and testing phase, the next step involved deploying the component using Docker containers. This deployment process was facilitated by creating a container that installed ROS along with all the necessary packages required to read Autoware messages. The Python component responsible for the CAM generation was executed within this container, ensuring seamless integration with the overall system. This containerized approach allowed for efficient deployment, easy scalability, and enhanced management of the component within the environment.

2) CPM: The Collective Perception Service (CPS) was introduced by ETSI to enable ITS-Ss to cooperate and exchange information about their surroundings, using local perception sensors. The aim of the CPS is to enhance overall awareness level on the roads, contributing to increased safety and efficiency for all road users. ETSI also defined the CPM which is the standard message to share information about objects on the road. CPMs are periodically transmitted with an adaptive frequency ranging from 1 Hz to 10 Hz. The CPM includes information about the current status of disseminating ITS-S (station type, speed, direction and others), its local perception sensors characteristics (sensor type, detection range and others), and information related to the perceived objects (object type, dynamic status and others). In order for a perceived object to be included in a CPM, a set of rules need to be met:

- It is the first detection of the object;
- Its absolute position changed by more than 4 m since

the last it was included in a CPM;

- Its absolute speed changed by more than 0.5 m/s since the last time it was included in a CPM;
- Its absolute velocity vector has changed by more than 4º since the last time it was included in a CPM;

The last time the object was included in a CPM was more than 1 second ago (0.5 seconds if the detected object is either from the class person or animal). In conclusion, the introduction of the Collective Perception Service is a step forward in the evolution of Intelligent Transportation Systems. By promoting cooperation and information exchange among ITS-Ss, the CPS greatly enhances road safety, optimizes traffic flow, and creates a smarter, more efficient transportation ecosystem for everyone involved. To generate a new CPM, a process similar to the generation of a CAM described previously was carried out. In the initial phase, the ROS messages exchanged in Autoware were examined to identify the topics containing the necessary information for the CPM as described earlier. Using the Autoware Simulator with a single moving vehicle, the information exchanged between different Autoware components was explored. After analyzing the message exchanges, it was determined that the topics $/awapi/vehicle/get/status$ and $/perception/object\_recognition/detection/rois0$ contained the required information for CPMs. The module created to subscribe to a ROS topic created and described in the previous section was used to obtain the data from Autoware. Additionally, a new class was created, encompassing all the fields required for CPM generation. An object of this class was instantiated to store the information received from the ROS topic. Simultaneously, at a frequency of 10Hz, this module initiated the CPM creation process. The object containing the CPM information is converted into a JSON representation of the message, which was subsequently published to a local MQTT broker located within the OBU to be encoded and transmitted by the NAP-Vanetza application. Upon completion of the development and testing phase, the Python component responsible for the CPM generation was successfully integrated and executed within the container created and described in the previous section.

3) DENM: Decentralized Environmental Notification Message (DENM) can be used by ITS-Ss to alert road users of a detected road event, such as a road accident or ongoing roadworks. These messages include information about the road event, namely its type, precise location, time of occurrence and validity. A new DENM generation process is triggered as soon as the road event is detected. The node itself, implemented in Python, would subscribe to the $vanetza/out/denm$ MQTT topic. Upon receiving a DENM , its content would be mirrored onto an equivalent ROS2 message structure, created for this application. Lastly, the structure would be published under the $v2x/denm/event$ ROS2 topic.

E. Brokering and integration

IV. Perception and Control

A. Perception service

B. Alert Service

An alert service has been developed to enhance road safety by detecting potentially dangerous situations and issuing warning messages based on camera information. Its architecture is despicted in Figure 2. The service is designed to identify events that trigger the generation of Decentralized Environmental Notification Message (DENM) - asynchronous messages used to communicate various road hazards. The events that can initiate the creation of a DENM include critical traffic conditions like traffic jams or stationary traffic and incidents of wrong-way driving. Additionally, in areas with high pedestrian-vehicle interaction, such as crosswalks, the presence of pedestrians can pose an increased risk of potential hazards. In the context of our specific scenario, a DENM is generated if a person is detected by the camera within a designated area, such as a crosswalk. This timely and proactive warning system is instrumental in fostering safer road environments, facilitating informed decision-making for drivers, and ultimately preventing accidents and improving overall road safety. The event detection module plays a crucial role in the alert service. It subscribes to relevant MQTT topics and diligently analyzes every received message to check for the presence of a road hazard event. Upon successful detection, a new DENM dissemination process is triggered. The DENM is created by utilizing a pre-existing JSON representation and then updating specific data elements related to the detected event. These updates include event detection time, and the event geolocalization in terms of latitude and longitude. Once the DENM is constructed, it is published to a specific MQTT topic that is actively monitored by NAP-Vanetza. The message is then encoded and transmitted to other ITS-Ss in the network, promptly alerting them to the potential danger.

C. Control Service

V. Scenario, Operation and Performance Evaluation

A. Devised Scenario

In active urban environments, ensuring road safety can be a complex challenge, particularly in crowded crosswalks where pedestrians and vehicles share the same space. The safety of both pedestrians and drivers becomes more crucial, especially when vehicles lack a direct line of sight to specific regions, making it challenging for their onboard sensors to detect potential hazards. To address this safety gap, a scenario where an innovative alert service, using a video camera, was explored. Positioned strategically at a busy crosswalk, an edge node continuously captures video data, acting as a watchful eye over the area. This real-time data is processed by the edge computing unit, evaluating pedestrian movements and identifying potential hazards. In this specific scenario, the alert service detects

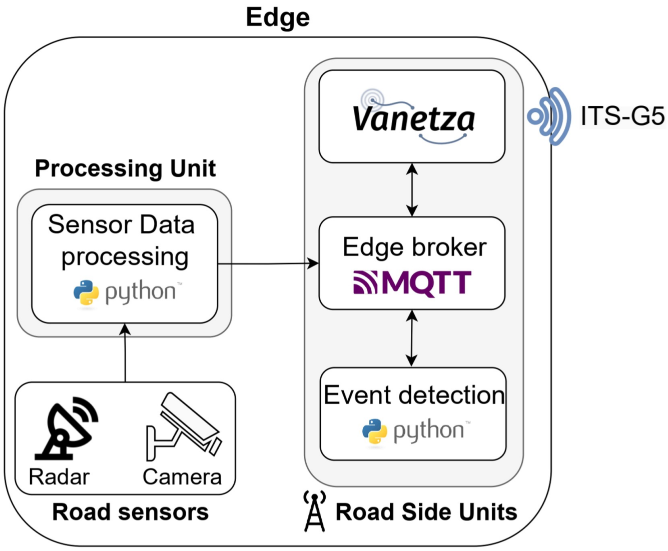

Fig. 2. Alert Service architecture

a pedestrian crossing the road, but due to obstructions or sensor limitations, a particular approaching vehicle cannot directly see the pedestrian. To address this visibility gap, the alert service takes immediate action by generating a Decentralized Environmental Notification Message (DENM). This asynchronous warning message is rapidly transmitted to nearby vehicles through a Roadside Unit (RSU), which acts as a communication hub, equipped with an ITS-G5 antenna. The RSU ensures that all approaching vehicles receive critical safety information regarding the presence of the pedestrian in the crosswalk. Through the collaborative efforts of multiple vehicular nodes, a robust safety network is formed. By effectively bridging the visibility gap between vehicles and the edge node's camera, the V2X communication system enhances overall road safety. This approach empowers vehicles to collectively perceive the road environment and make informed decisions, proactively mitigating potential collisions and enhancing road safety.

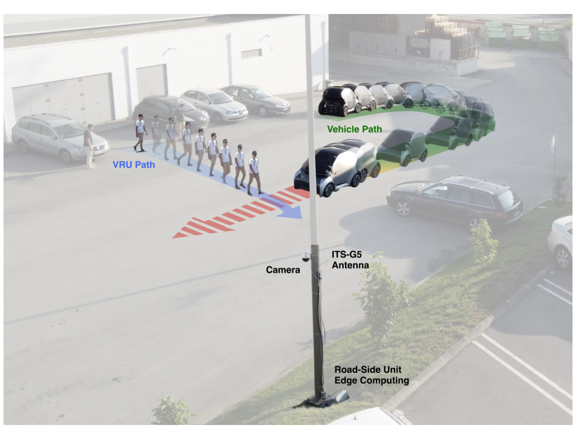

Fig. 3. scenario.

B. Setup and Experimental Platform

1) Edge Node: Edge nodes serve as versatile platforms that combine various functionalities to support efficient V2X communication. These nodes encompass communication access points, Multi-access edge computing (MEC) capabilities, and sensing devices, all contributing to the seamless exchange of data in the V2X ecosystem. At the heart of enabling efficient V2X communication lies the RSU, a central player in the network. Our implementation utilizes a PC Engines single-board computer (SBC), equipped with a mini-PCIe wireless card operating on the 5.9 GHz band. This mini-PCIe wireless card acts as a crucial enabler for ITS-G5 Vehicle to Everything (V2X) communications, facilitating reliable and data exchange between vehicles and infrastructure. Moreover, the RSU can be equipped with an additional wireless card to function as a WiFi access point (AP), providing connectivity services to nearby devices. For comprehensive perception and situational awareness, the edge node is equipped with a single video camera. This integrated camera serves as an essential sensing device, capturing real-time data from the surroundings. This data is crucial for detecting potential road hazards and triggering timely responses to ensure road safety. To handle the processing demands of the collected data, the edge node is equipped with an Nvidia Jetson Nano, boasting a powerful GPU for graphic-intensive processing. This MEC component enables rapid and efficient data processing, empowering the edge node to analyze the camera data in real-time. By leveraging the Jetson Nano's capabilities, the edge node can quickly extract valuable insights from the video feed and make informed decisions to enhance the overall efficiency of the V2X system. To provide a comprehensive visual representation of the edge node's perceptions, the RSU is connected to the Aveiro Tech City Living Lab (ATCLL) dashboard. This connection is established through an internet connection available at the test site. The dashboard serves as a powerful tool for gathering and displaying the messages exchanged within the RSU. By leveraging this internet connection, the dashboard continuously collects and analyzes the messages transmitted and received by the RSU. It acts as a central hub that captures the data exchanged in real-time. The ATCLL dashboard offers a user-friendly interface, presenting the gathered data in an easily understandable and visually appealing format. Users can access this dashboard through their mobile devices to observe the V2X communication activities and the edge node's perceptions. By monitoring the messages exchanged within the RSU, the dashboard offers a unique and valuable perspective on the dynamic interactions between vehicles and the edge node.

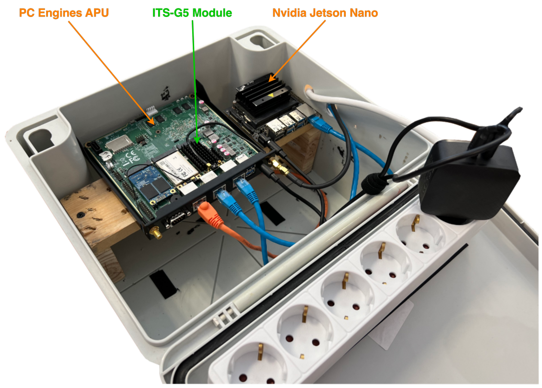

Fig. 4. Edge Node.

C. Operation Results

D. Performance Results

VI. Conclusions and Future Work

References

[1] Y. Asabe, E. Javanmardi, J. Nakazato, M. Tsukada, and H. Esaki, "Autowarev2x: Reliable v2x communication and collective perception for autonomous driving," 06 2023.

[2] T.-K. Lee, T.-W. Wang, W.-X. Wu, Y.-C. Kuo, S.-H. Huang, G.-S. Wang, C.-Y. Lin, J.-J. Chen, and Y.-C. Tseng, "Building a v2x simulation framework for future autonomous driving," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019, pp. 1–6.