MuSLi: a Multi Sensor LiDAR Detection for C-V2X Networks

Romeo Giuliano, Anna Maria Vegni, Valeria Loscrì, Eros Innocenti, Alessandro Vizzarri, Franco Mazzenga

MuSLi: a Multi Sensor LiDAR Detection for C-V2X Networks

Romeo Giuliano[a], Anna Maria Vegni[b,*], Valeria Loscrí[c], Eros Innocenti[a], Alessandro Vizzarri[d], Franco Mazzenga[d]

[a]*Department of Engineering Science, Guglielmo Marconi University, Italy*
[b]*Department of Industrial, Electronic, and Mechanical Engineering, Roma Tre University, Italy*
[c]*INRIA Lille-Nord Europe, France*
[d]*Department of Enterprise Engineering, University of Rome Tor Vergata, Italy.*

Abstract

Obstacle detection is a tool adopted in vehicular safety applications, aiming to detect a moving obstacle in an area of interest with the highest accuracy. Different sensors are used for this aim, such as LiDAR devices mounted on board of a vehicle that capture images of the surrounding environment. Extending the number of LiDAR sensors can be useful to improve the obstacle detection accuracy, since multiple images are captured from different distances and directions, and this represents an interesting approach, specially in case of dense networks with cooperative nodes. In this paper we present MuSLi technique, aiming to (*i*) provide an accurate obstacle detection and (*ii*) forward alert messages to other cars in the network, in case of correct detection of a pedestrian crossing the street. MuSLi relies on the connected content islands scenario, where each vehicle defined as a content island subscribes to a service in order to receive and share published messages. Specifically, the road safety service allows the detection of an obstacle through multiple LiDAR sensors from neighboring cars. Furthermore, we investigate the fastest transmission mode among those defined in the C-V2X releases to alert the presence of a pedestrian to the other approaching cars. The proposed technique provides the distances by the crossroad in which is better to use V2V, V2I or V2N according to the environment, the scheduling technique and the measured interference.

Keywords: Cellular-Vehicle to everything (C-V2X), Publish/Subscribe,

*Corresponding author

Email addresses: r.giuliano@unimarconi.it (Romeo Giuliano), annamaria.vegni@uniroma3.it (Anna Maria Vegni), valeria.loscri@inria.fr (Valeria Loscrí), eros@newtechweb.it (Eros Innocenti), alessandro.vizzarri@uniroma2.it (Alessandro Vizzarri), mazzenga@ing.uniroma2.it (Franco Mazzenga)

Machine Learning, Multi-sensor detection.

1. Introduction

In the near future, it is envisioned that with 6G we will assist at a new era of billions of things, drones, robots, people and vehicles connected to each other and generating Zettabytes of digital information [1]. In this context, novel services and applications will be more challenging, and enriched with immersive communication and characterized with more stringent requirements. In this sense, 6G is regarded as the key future generation network, able to tackle these challenges, by encompassing both human and machine-centric philosophy [2].

In the vehicular context, 6G is regarded as the technology allowing the development of simultaneous localization and mapping (SLAM) approaches, for enabling advanced cross reality applications encompassing autonomous vehicles and drones. In [3], Huang *et al.* provide a detailed survey on SLAM. In particular, they consider Light Detection And Ranging (LiDAR) and Visual SLAM and describe the devices, their evolution, the main challenges and finally provide a constructive discussion on future directions. With the deployment of 5G around the world, research community and industry have already started to investigate the 6G network, and specifically the concept of *network intelligentization*, that will replace the virtualization concept that characterized 5G networks [4]. Intelligentization is meant as connected intelligence with massive use of Machine Learning (ML) capability for enabling the capacity of 6G systems to embrace new emerging technologies, such as THz and intelligent reprogrammable surfaces, thus enabling a reprogrammable environment paradigm. The key factors in such a kind of complex environment are represented by the massive use of Artificial Intelligence (AI) and the high heterogeneity and complexity of the 6G systems, where the softwarization concept, featured in 5G networks, will not be enough to support the new requirements and applications [5].

In the context of Internet of Vehicles (IoV) paradigm, as part of the Intelligent Transport System (ITS) and with the advent of autonomous vehicles,

one of the most important functionality to be developed is the obstacle detection [6, 7]. In order to develop advanced control systems for detecting obstacles, a combination of advanced sensors, radar and camera need to be considered for building an accurate environment perception. A first categorization of sensors/devices for obstacle detection considers cooperative and non-cooperative sensors, as described in [8]. Among the different cooperative sensors, we remind the Automatic Dependent Surveillance and Broadcasting (ADS-B) and the Traffic Alert Collision Avoidance System (TCAS), while as non-cooperative sensors, we have infrared, electro-optical, acoustic, LiDAR, sonar, radar, etc.

In [9], Habermann and Garcia focus on laser sensors for realizing obstacle detection. LiDAR technology is introduced in [10] for collision avoidance, and the authors apply a fusion technique to 3D LiDAR and 2D image data for realizing efficient anti-collision management. The LiDAR sensor and a camera sensor are exploited for obstacle detection; the fusion of the images is then realized, in order to improve the accuracy of the single image and an improvement of the latency is also obtained based on this fusion algorithm. The use of LiDAR for obstacle detection has been employed also in [11], where line segments are extracted by processing raw data of the ranging sensor and downward-looking light detection. This approach allows a good distinction between line segments and obstacles under various road conditions.

Anyway, in the most of papers from the literature, the analyzed sensors and the proposed fusion techniques foresee that sensors are mounted on board on the same vehicle, with the constraint to have a limited –single– vision of the investigated area. On the other side, by considering a networked technique where multiple nodes can cooperate in a joint manner for a common task, *e.g.* the obstacle detection, we expect to obtain an enhanced vision of the investigated area, and then the obstacle detection will be likely more accurate as compared to single sensor-based approaches.

Leveraging on current literature results, in this paper we aim to investigate a multi-sensor obstacle detection approach for fast and accurate detection, to be adopted in safety vehicular applications. Specifically, we adopt ML techniques

for pedestrian detection in vehicular scenarios through the use of multiple LiDAR sensors mounted on vehicles. The pedestrian detection task occurs through the collection of multiple signals coming from the available neighboring vehicles, that are approaching an area of interest *e.g.*, a crossroad. The fusion of collected LiDAR images (*i.e.*, clouds of points) is useful to increase the accuracy of obstacle detection; as a result, the probability of detection will result increased and more accurate.

1.1. Main contributions

In this paper, we present a multi sensor LiDAR detection technique, namely MuSLi, aiming to detect obstacles in a vehicular environment. We extend the "content island" model, as introduced in [12], so that both vehicles (*i.e.*, cars [1] and Road Side Units (RSUs) are equipped with various sensors and data capture devices, and are able to exchange data to neighboring vehicles and also process them. Specifically, content islands are a set of interconnected vehicles and RSUs with sensing and processing capabilities, which exchange data using a publish/subscribe system. Exchanged messages can either contain data or an action trigger, which is a request originating from one vehicle/RSU soliciting approaching nodes to process a dataset or simply inquiring for a given query. For this aim, we can rely on the Message Queue Telemetry Transport (MQTT) protocol to provide the required publish/subscribe-based communication. MQTT is a widespread standard in Internet of Things (IoT) for publish/subscribe systems, and in the context of vehicular networks, it is being used in a wide range of applications such as to provide collision warnings [12], to design a traffic control system to minimize the response time of emergency vehicles [13], and for sharing energy between two electric vehicles [14].

Thanks to the LiDAR sensors mounted on board, "content island" cars close to a crossroad are able to detect pedestrians crossing the street. They share their collected LiDAR clouds in order to perform a detection based on multiple point

[1] The terms "vehicle" and "car" are used interchangeable in this paper.)

clouds for the same pedestrians, and thus improving the detection accuracy. As compared to single-sensor techniques, MuSLi will provide higher obstacle detection probability, reaching an enhancement of 35% of accuracy at a distance of 10 m from the area of interest. As expected, the higher the number of multiple LiDAR point clouds, the higher will be the detection probability.

Notice that MuSLi approach has been intended with the twofold goal of (*i*) obstacle detection, and (*ii*) alert message forwarding. Indeed, after detecting a pedestrian crossing the street, this information triggers a message forwarding task, where cars will disseminate alerting messages to other approaching cars through Cellular-Vehicle to everything (C-V2X) approach [15]. Different V2X scenarios require the transport of messages with different performance requirements for the 3GPP system. In [16], Extended Sensors is one of the six different use cases with 3GPP support for V2X scenarios. It enables the exchange of raw or processed data gathered through local sensors or live video data among cars, RSUs, devices of pedestrians and V2X application servers. In the proposed framework, cars approaching to a given area (*e.g.*, a crossroad) may subscribe to the extended sensor service. They are able to publish the contents collected by their on-board sensors (*i.e.*, the LiDAR clouds) and receive data from the other subscribed cars. As a result, the awareness of the road environment is enhanced due to data from multiple sensors. In case of obstacle detection *e.g.*, a pedestrian crossing the crossroad, subscribers can send alarm messages to approaching vehicles. In this context, we also evaluate the delay in message alarm delivery by considering a dynamic and a semi-persistent scheduling strategies, for specific propagation environments such as urban, sub-urban and rural, transmission modes *i.e.*, Line-Of-Sight (LoS) and Non-Line-Of-Sight (NLoS), and the amount of interference power in the considered area. The main objectives of this paper are as follows:

1. We introduce MuSLi technique, aiming to detect obstacles in vehicular environments. A source vehicle, defined also as content island, exploits the collection of multiple LiDAR point clouds coming from neighboring

cars, close to an area of interest (*e.g.*, a crossroad), which are able to detect an obstacle thanks to their LiDAR sensors installed on board. A pool of signals from multiple LiDAR sensors are collected and processed by means of a ML approach, for a more accurate obstacle detection, as compared to single-sensor approaches;

2. MuSLi is intended with the twofold aim of both (*i*) obstacle detection and (*ii*) alert message forwarding. The latter task is triggered by the previous one, in case of positive obstacle detection. Alert messages are then forwarded to approaching cars, by means of C-V2X. We will evaluate different communication strategies on some propagation environments and signal interference levels, in order to select the optimal communication mode that can guarantee the fastest message forwarding;
3. MuSLi runs at application level as in [17] and allows data exchange among cars according to a typical distribution protocol such as MQTT. The performance assessment of MuSLi is carried out by means of simulation results expressed as both the accuracy of obstacle detection and the latency in transmitting alert messages via C-V2X. According to the environment, propagation conditions and distances from the approaching cars, we will discuss about the optimal transmission method to provide a reduced latency in message delivery.

This paper is organized as follows. In Section 2 we present the system model adopted to represent mathematically the reference scenario. We detail the formulation of the MuSLi performance, expressed in terms of outage probability and latency in different communication and transmission modes. An optimization problem related to the latency on transmission links is also addressed. Our proposed MuSLi technique is presented in Section 3, where we first introduce how obstacle detection occurs through an ML approach, and then how to forward alert messages in case an obstacle is detected. A more detailed description of how MuSLi can detect an obstacle is presented in Section 4. The effectiveness of MuSLi is then assessed in Section 5 for different communication modes and

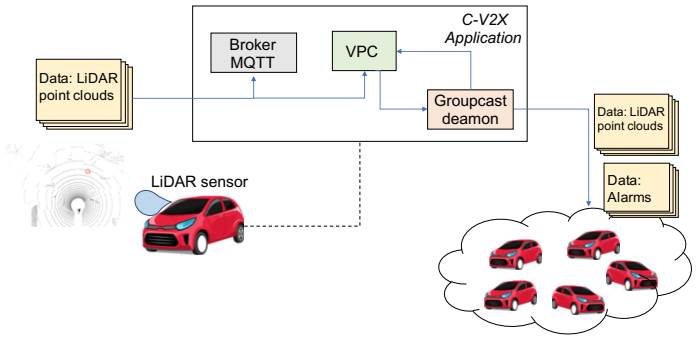


Figure 1: Schematic of content island vehicle, relying on MQTT for message exchange.

vehicular scenarios. Finally, conclusions are drawn at the end of the paper.

2. System Model

A short description of the reference scenario and the different communication modes, providing different network performance, is introduced hereafter.

2.1. Reference Scenario

In the following, we report the scenario and the modeling of the reference system. We consider that one or more cars are close to a crossroad. Each car, defined as *content island*, is equipped with a LiDAR sensor able to scan the crossroad area and to detect possible pedestrians crossing the neighboring streets. Based on the detection performed by cars close to the crossroad, alerting messages are sent to other cars approaching the crossroad, with the aim of signaling the presence of pedestrians (hard to be seen by the drivers due to the distance), thus allowing them to properly adapt their speed. Message exchange and forwarding occur by means of a publish/subscribe system. For instance, a node can be subscribed to a service *i.e.*, road safety, and receive a published message *i.e.*, a warning alert related to obstacle detection, from another node in a given area. Similarly, a source node collects different information from neighboring cars *i.e.*, LiDAR images, process them and finally publish the obtained result *i.e.*, obstacle detection.

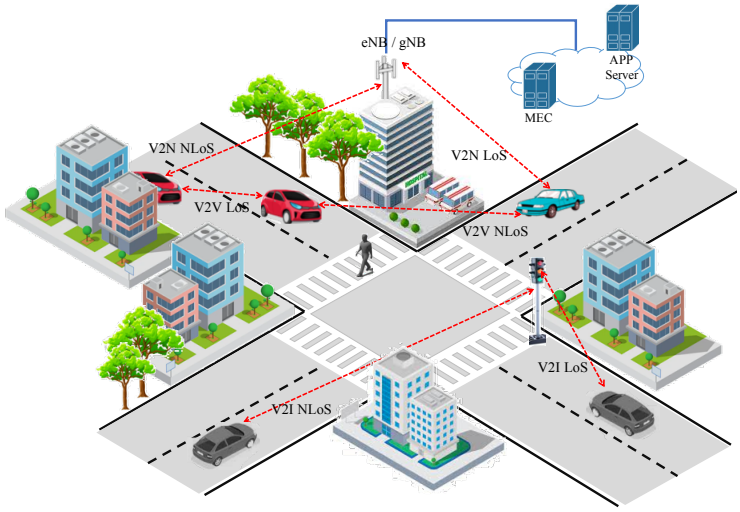


Figure 2: Connected content islands reference scenario, where different communication and transmission modes are allowed.

The content island cars are based on the model introduced in [12, 18]. In this paper, we consider the island core is comprised of an MQTT broker, a Virtual Processing Client (VPC), and the multicast daemon, as well depicted in Figure 1. The MQTT broker is the element to which LiDAR sensor present inside a vehicle is connected. By connecting to the broker, it may publish information that is delivered to other cars having subscribed to such a service. A VPC is a special client aims to control the flow of data information, while the groupcast daemon is a service used by the VPC to send and receive broadcast message from neighboring content island cars. For this purpose, VPC sets up MQTT messages to connect multiple content islands for data exchange.

After introducing the content island vehicle concept, in Figure 2 we report the elements envisaged in a "Connected Content Islands" scenario. In addition to the cars able to transmit messages among them at short distances, RSUs are present along the roads and able to directly communicate to the cars in proximity. Moreover, it is possible to provide communication among cars by using local deployed base stations (*i.e.*, eNB and gNB). From Figure 2, the following vehicular communication modes are highlighted:

- Vehicle-to-Vehicle (V2V), in case the communication occurs between two On-Board Units (OBUs) mounted on the content island cars;
- Vehicle-to-Infrastructure (V2I), in case the communication occurs between an OBU and a content island RSU;
- Vehicle-to-Network (V2N), in case of communications among content island cars through a local base station over the cellular network, thus enlarging its reachable area. In this case, we observe the communication can occur through a Cloud server or a Mobile/Multi-access Edge Computing (MEC) node, which relay the message.

Notice that all these communication modes are possible for both LoS and NLoS transmission links, due to the presence of obstacles (*e.g.*, buildings, trees, etc.) along the transmission range. In the next section, we will observe that the most appropriate communication mode depends on channel conditions and specific scenario characteristics that affect the signal propagation.

2.2. Evaluation Model

In order to assess the performance of the considered system, we evaluate the outage probability for different transmission modes both in downlink and uplink[2], which indicates the probability that a message sent by a vehicle is correctly received by a destination node *i.e.*,

$$P_{out} = \Pr\left\{SINR < \rho_0\right\}, \tag{1}$$

where ρ_0 is the Signal-to-Interference plus Noise (SINR) threshold for the considered service. In Eq. (1), the SINR parameter can be expressed as:

$$SINR = \frac{P_T \cdot G_T G_R}{L(d)} \cdot \left(\frac{1}{\eta \cdot \left(1 + \frac{I}{\eta}\right)} \right), \tag{2}$$

where P_T [W] is the transmitting power, G_T and G_R are the antenna gains for the transmitter and the receiver respectively, $L(d)$ is the pathloss accounting for

[2]We assume that downlink and uplink are balanced.

Table 1: Main parameters adopted in different environments and communication modes.

V2V	**V2I**	**V2N**		
$h_{veh} = 1.5$ m	$h_{BS} = 4$ m	$h_{BS} = 10$ m		
$G_{T/R} = 3$ dB	$G_{T/R} = 3$ dB	UMi	UMa	Rural
		$G_{T/R} = 10$ dB	$G_{T/R} = 15$ dB	$G_{T/R} = 18$ dB

the losses due to the distance d [m] between the transmitter and the receiver, and η [W] is the thermal noise power. The ratio I/η is the noise raise caused by the extra interference due to other simultaneous communications occurring in the same Physical Resource Blocks (PRBs).

In Eq. (2), the expression of pathloss depends according to different scenarios and transmission modes. Specifically, from [19], we have considered three different environments *i.e.*, (*i*) Urban Micro, (UMi), (*ii*) Urban Macro, (UMa), and (*iii*) Rural, and the radio transmission in case of (*i*) LoS and (*ii*) NLoS. Only the slow fading has been taken into account, since the small-scale fading is compensated in the Orthogonal Frequency-Division Multiplexing (OFDM) receiver by the one tap equalizer. Table 1 collects the parameters adopted for the computation of pathloss in case of different environments and transmission modes, where h_{veh} [m] and h_{BS} [m] are the heights of the vehicle and of the Base Station (BS), in case of V2V, V2I and V2N, respectively. For the results in Section 5, we considered $P_T = 20$ dBm and $\rho_0 = 15$ dB [20, 21].

Specifically, we have considered that in Urban Micro (*i.e.*, street canyon, open area) the BSs are mounted below rooftop levels of surrounding buildings. Open area is intended to real-life scenarios capture, such as a city or station square. On the other side, in Urban Macro, the BSs are mounted above rooftop levels of surrounding buildings, while the rural deployment scenario focuses on larger and continuous coverage. The key characteristics of this scenario are continuous wide area coverage supporting high speed cars.

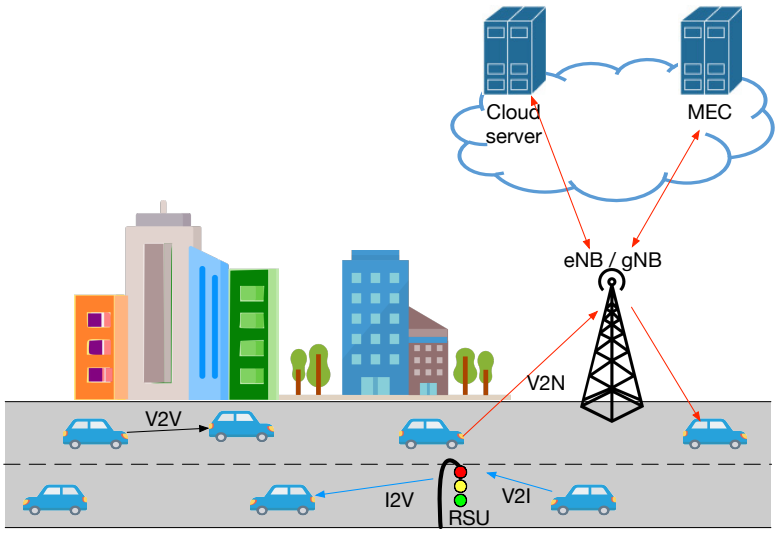


Figure 3: Vehicular communication modes in the considered scenario.

2.3. Latency assessment

According to the schematic in Figure 3, we considered different vehicular communication modes *i.e.*, V2V, V2I, and V2N, and for each of them, the communication involves different elements in the reference network. In V2V mode, the communication occurs directly between two neighboring cars in LoS, and no further element is considered (*black arrow* in Figure 3). In V2I, a vehicle approaching to a RSU transmits a message to it (*i.e.*, *blue arrow* of V2I connectivity link), which relays it to another approaching vehicle (*i.e.*, *blue arrow* of I2V connectivity link). Finally, in V2N mode a vehicle communicates with a local Multi-Access Edge Computing (MEC) or a remote Cloud server through the local eNB/gNB, and then the message will be redirected toward a destination vehicle (see *red arrows* in Figure 3). In this case the destination vehicle will receive the alerting message through the transmission link from the eNB/gNB, after the processing performed by the MEC or by the remote Cloud in the network. In Figure 3 communication paths of the three communication modes are highlighted in black, blue and red lines, for V2V, V2I and V2N, respectively.

Table 2: Latency ranges [ms] in case of (i) Dynamic and (ii) Semi-persistent scheduling.

Parameter	LTE/5G @15kHz	5G @60kHz	5G Mini-slot	V2I	V2V
Dynamic scheduling					
Frame alignment	$[3, 6]$	$[3, 6]$	$[0.5, 1]$	$[0.5, 1.5]$	$[0.5, 1.5]$
PHY transmission	$[15, 18]$	$[4, 5]$	$[0.3, 0.8]$	$[9, 11]$	$[9, 11]$
eNB/gNB - MEC/GW	$[2, 5]$	$[2, 5]$	$[4, 6]$	-	-
GW-Cloud	$[3, 5]$	$[3, 5]$	$[4, 6]$	-	-
Semi-Persistent Scheduling (SPS)					
Frame alignment	$[3, 5]$	$[3, 5]$	$[0.5, 1]$	$[0.5, 1.5]$	$[0.5, 1.5]$
PHY transmission	$[5, 8]$	$[1, 2]$	$[0.3, 0.8]$	$[1, 2]$	$[1, 2]$
eNB/gNB - MEC/GW	$[2, 5]$	$[2, 5]$	$[4, 6]$	-	-
GW-Cloud	$[3, 5]$	$[3, 5]$	$[4, 6]$	-	-

In order to assess the latencies for each scenario and in each environment, in Table 2 we report the range of values, elaborated from [22] and [23]. In our analysis, in addition to V2I and V2V, for V2N we considered (i) two sub-carrier spacing (SCS) values of 15 kHz and 60 kHz, with a slot duration of 1 ms and 0.125 ms, respectively, and (ii) a mini-slot case with 7 OFDM symbols. Moreover, we considered two types of scheduling *i.e.*, (i) Semi-Persistent Scheduling (SPS) and (ii) Dynamic scheduling [24]. In the first case, we assume that the Physical Resource Block (PRB) is already scheduled and no contention has to be performed by the transmitting OBU. In the second case, before transmitting, the OBU has to send a scheduling request message, then waits for the PRB assignment through the Downlink Control Information (DCI) and finally, transmits the alert message [24]. Based on data collected in Table 2, in Figure 4 we show the multiple contributions for the computation of the latency Cumulative Distribution Function (CDF), computed for each communication mode (*i.e.*, V2V, V2I, and V2N), in case of dynamic and semi-persistent scheduling,

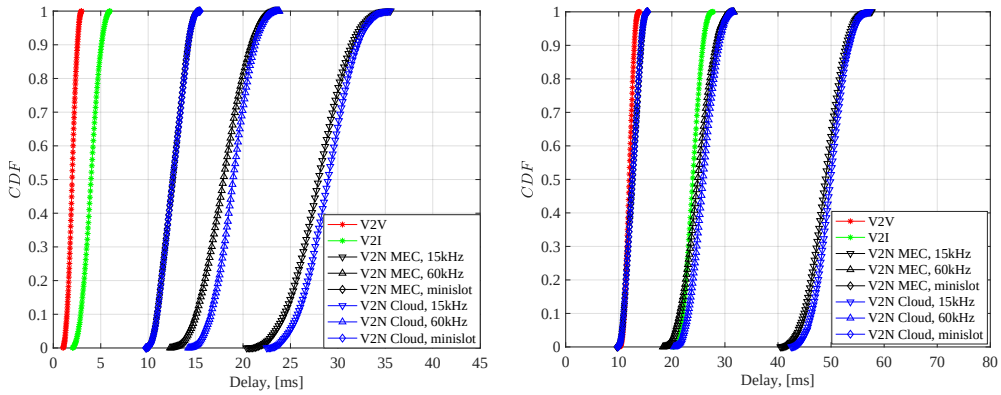


Figure 4: Comparison of the latency CDF in case (a) semi-persistent and (b) dynamic scheduling, respectively.

and assuming no packet loss. Specifically, we consider the latency of physical transmission as the sum of different latencies such as the downlink and uplink transmission delay, the downlink and uplink slot alignment, and the UE and BS processing.

However, in a real environment, packet loss can occur, thus increasing the delay in communications between connected cars approaching to the crossroad. It depends on the number of retransmissions and on the outage probability. Then, for each communication mode *i.e.*, V2V, V2I and V2N, and for each environment *i.e.*, UMi, UMa, and rural, the number of retransmissions varies based on the distance between (i) two cars, (ii) the approaching car and the RSU, and (iii) the car and the eNB/gNB, respectively, as well as the interference power. We can define the overall latency $\mathcal{L}$ [ms] as:

$$\mathcal{L}_m = N_{tx,avg}\left(P_{out}\right) \cdot \delta_m(CDF \geq \chi), \tag{3}$$

where δ_m [ms] is the delay occurring for a single alert message successful transmission, which depends on the m-th transmission mode *i.e.*, $m = [1, 2, 3]$, corresponding to V2V, V2I, and V2N, respectively, and is associated to the latency CDF, which should be higher than a given threshold χ. For design purpose, it

is possible to consider the average or any other percentile value of the latency CDF, which is depicted in Figure 4 (*e.g.*, the 90-th percentile or 99-th percentile). In Eq. (3), $N_{tx,avg}$ is the average number of transmissions necessary to deliver an alert message to a destination node, based on the outage probability P_{out} obtained for a given scenario, which is equal to:

$$N_{tx,avg} = \frac{1}{M} \sum_{i=1}^{M} N_{tx,i}(p_i < P_{out}), \tag{4}$$

where $N_{tx,i}$ is the number of transmissions necessary to deliver the i-th alert message, depending on a given transmission probability p_i that should be lower than the outage probability associated to a given scenario and communication mode, and M is the number of messages to be delivered. Finally, MuSLi will select the optimal transmission mode (*i.e.*, $m^{\star}$) defined as:

$$m^{\star} = \arg\min \mathcal{L}_m. \tag{5}$$

3. MuSLi technique

The main tasks of MuSLi approach are detailed in Algorithm 1. MuSLi is a technique for (i) fast and accurate obstacle detection and (ii) resulting alert message forwarding. For the obstacle detection, multiple cars are involved in this task, so that multiple LiDAR point clouds are collected and processed in order to compute a joint detection. If the detection score is higher than a given threshold, MuSLi will transmit alert messages related to the presence of an obstacle in a given position on the road. Messages transmission occurs through C-V2X communications, in case of V2V, V2I and V2N modes, and different transmissions links, such as LoS and NLoS.

Let us assume that in a given reference environment *i.e.*, Urban Micro, Urban Macro, and Rural, a source vehicle *i.e.*, V_{Tx}, is approaching to an area of interest where it is likely to find an obstacle (*e.g.*, a pedestrian crossing the street). It is able to communicate with its neighboring vehicles *i.e.*, $(v-1)$ vehicles, according to the m-th (with $m = [1, 2, 3]$) communication modes (*i.e.*, V2V, V2I, and V2N, corresponding to $m = 1$, $m = 2$, and $m = 3$, respectively). The source vehicle

and the other $(v-1)$ vehicles are equipped with LiDAR sensors and periodically scan the environment acquiring clouds of LiDAR points. All these acquired data are exchanged among vehicles and the cumulated LiDAR cloud resulting from the sum of collected clouds by the other vehicles is the input for an on board neural network, detailed in Section 4, with the aim of pedestrian detection. As a result, MuSLi will provide information about the presence of finding an obstacle at distance d [m] from the source vehicle (or in general from a d_{cross} distance from the area of interest).

Let $n = [1, 2, \ldots, N_d]$ be the index of the output from the neural network, related to the obstacle detection decision. We assume that the n-th Boolean variable $\mathcal{O}_n$ represents the single information about obstacle detected (*i.e.*, $\mathcal{O}_n = 1$) or not (*i.e.*, $\mathcal{O}_n = 0$) in case the n-th output of the neural network (*i.e.*, the confidence score D_n) is higher than a given threshold D_{th} (*i.e.*, $D \geq D_{th}$). In case that the majority of the N_d outputs is positive, MuSLi will consider the obstacle as present and detected *i.e.*, $\mathcal{O} = 1$, and will act for message forwarding from the source vehicle to other approaching vehicles (*i.e.*, N_v). On the other side, the obstacle is assumed as not present and will be discarded *i.e.*, $\mathcal{O} = 0$. This is mathematically expressed as:

$$\mathcal{O} = \begin{cases} 1 & if \ \sum_{n=1}^{N_d} \mathcal{O}_n < \lceil N_d/2 \rceil + 1 \\ 0 & otherwise \end{cases} \tag{6}$$

Based on the given reference environment, the function $Alert\_message\_forwarding()$ will compute the outage probability P_{out} towards the i-th recipient vehicle and the expected latency for the m-th communication modes (*i.e.*, $\mathcal{L}_m$). An optimization of the latency will occur, so that the source vehicle will be forced to transmit an alert message through the m-th communication mode that provides the lowest latency *i.e.*, $m^\star$. More details about the mathematical modeling of the MuSLi technique are provided in next Section 4. The proposed algorithm works under the constraint of a maximum allowable number of retransmissions of warning messages *i.e.*, $N_{\max,tx}$. This parameter has been introduced in order to limit the latency in case of high values of outage probability that can affect

Algorithm 1: MuSLi Pseudocode

Input : V_{Tx} ▷ Source vehicle
v ▷ Available vehicles for obstacle detection
$p_c^{(\ell)}$ ▷ Point cloud from the ℓ-th vehicle
D_{th} ▷ Threshold for obstacle detection
n ▷ Index of decision
N_d ▷ Number of decisions
r ▷ Index of retransmissions
$N_{\max,tx}$ ▷ Maximum number of retransmissions
Output: $\mathcal{O}$ ▷ Boolean variable for obstacle detection

foreach $n = [1, N_d]$ **do**
  foreach $p_c^{(\ell)}$, $\ell = [1, v]$ **do**
    compute P_c ▷ Eq. (7)
    P_c enters the detector
    compute D ▷ Detection score
    if $D \geq D_{th}$ **then**
      $\mathcal{O}_n = 1$ ▷ Single output of detected obstacle
    else
      $\mathcal{O}_n = 0$ ▷ Single output of obstacle not detected

if $\mathcal{O} = \left(\sum_{n=1}^{N_d} \mathcal{O}_n\right) \geq \lceil N_d/2 \rceil + 1$ **then**
  Input : $\mathcal{S}$ ▷ Reference environment
    m ▷ Index of transmission mode
    N_v ▷ Number of vehicles for forwarding
    $\mathcal{D}$ ▷ Set of V_{Tx}-N_v distances
  Output: $m^\star$ ▷ Optimal transmission mode
  $Alert\_message\_forwarding(\mathcal{S}, m)$
  foreach $i = [1, N_v]$ **do**
    compute $P_{out}(d_i)$ with $d_i \in \mathcal{D}$ ▷ Outage probability
    compute $\mathcal{L}_m$ ▷ Latency for V2V, V2I and V2N
    $\min_{m=[1,3]} \mathcal{L}_m$ ▷ Optimal transmission mode selection
    $m^\star : \mathcal{L}_{m^\star} = \min_{m=[1,3]} \mathcal{L}_m$ ▷ Compute $m^\star$
    while $r < N_{\max,tx}$, $r = [1, N_{\max,tx}]$ **do**
      send message to the i-th vehicle by $m^\star$
      if *successful transmission* **then**
        exit
      $r + 1 \leftarrow r$

else
  Obstacle is not detected and will be discarded

the success of message delivery. In case of an error occurs in the r-th transmission of a warning message, with $r = [1, \ldots, N_{\max,tx}]$, the vehicle that detected the crossing pedestrian will try again to send a message until a maximum number of transmission *i.e.*, $N_{\max,tx}$. When the maximum number of attempts is reached *i.e.*, $r \rightarrow N_{\max,tx}$, the vehicle stops sending warning messages and no info is forwarded to the approaching vehicle, then achieving an infinity latency.

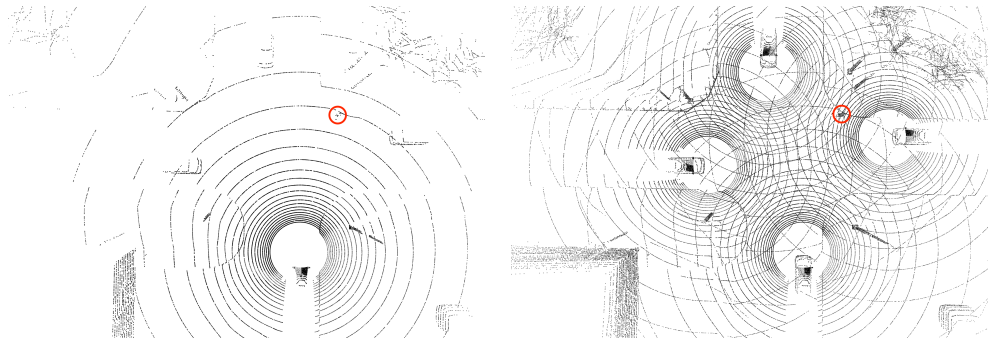
In case of obstacle detection but the obstacle is not present (*i.e.*, false positive), the MuSLi algorithm still sends the warning messages. Similarly, if the obstacle detection is negative but the obstacle is present (*i.e.*, false negative), MuSLi does not send the warning messages. In order to limit these flaws, we introduced the possibility to consider multiple outputs of the neural network (*i.e.*, $\mathcal{O}_n$) in sequence from the neural network and take a decision on the pedestrian detection on a majority basis.

In order to improve the detection of the on board-mounted LiDAR, we assume that the vehicles close to the crossroad collect the LiDAR traces of the same scene autonomously, but from different points of view. Then, they share each own LiDAR trace with the other vehicles. Each vehicle is able to superimpose its own LiDAR trace with the other received traces, by properly translating and rotating them to synchronize them with its collected trace. Now, the neural network is able to provide whether (and where) a pedestrian is crossing the street. The superimposed traces increase the accuracy of the LiDAR detection, with respect to a single trace. An example of a single and a superimposed LiDAR point clouds is reported in Figure 5.

4. Multi-LiDAR Sensor Detection

4.1. Testbed for Evaluation

In order to evaluate the combination of multiple points of clouds from LiDAR sensors mounted on vehicles, we used the open-source simulator CARLA [25] for autonomous driving to implement the dedicated testbed. The flexibility and degree of realism offered by this simulator allowed creating complex environments,



(a) (b)

Figure 5: Example of detection image obtained from (a) one single point cloud of a LiDAR sensor mounted on board of one vehicle, and (b) four superimposed clouds coming from multiple LiDAR sensors mounted on four different vehicles. The pedestrian detection is highlighted in the red circle.

which form the basis for precise and reproducible experiments. We implemented the scenario depicted in Figure 6, using the bundled map Town07. The main scene is composed by 4 vehicles actors, each one equipped with a LiDAR sensor.

The simulated LiDAR characteristics are analogue to the Velodyne HDL32E device *i.e.*, 20 Hz capture frequency, 32 channels, 360° horizontal Field of View (FOV), $+10°$ to $-30°$ vertical FOV, $[80, 100]$ m range with usable returns up to 70 m, ≈ 1.39 million points per second. Our simulation framework gives the possibility to create snapshots of the environment that contain actors position, synchronized reading from every sensors (*e.g.*, LiDAR, GNSS, mounted cameras, Radar and so on) at a specific moment of the simulated time.

In order to enforce the reproducibility of our experiments, we configured the underlying rendering engine to perform in synchronous mode. Using this method, the rendering engine processes a single step and then it waits for the client permission to go ahead with another. We chose a single step duration equal to 50 ms of simulated time, thus obtaining 20 steps for 1 second. In this experiment, we started by spawning a pedestrian willing to cross the road

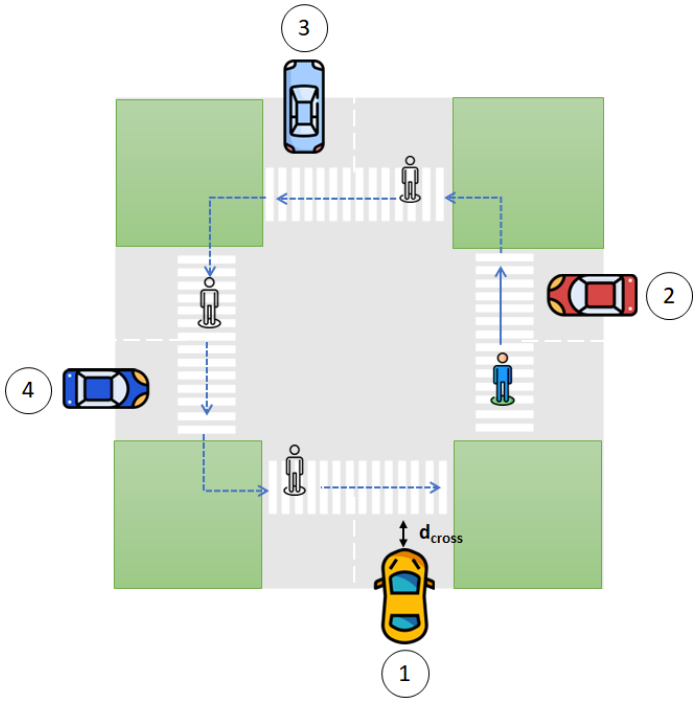


Figure 6: Simulated scenario for obstacle detection, where one pedestrian is crossing the area along the four pedestrian crossings.

with a constant speed of 1.3 m/s at the east intersection side. After that, we collected several snapshots of the environment, consisting in 40 LiDAR point clouds coming from the car denoted by number 1 in Figure 6. The simulation continued by moving the car backwards of a distance of d_{cross} at steps of 5 m from the crossroad, resetting the walker to the initial position, and restarting the environment snapshot collection from the beginning. Eventually, we repeated the entire process by adding more vehicles (*i.e.*, car 2, 3 and 4) to the detection system, and comparing the obtained detection performances between the single car scenario versus the multiple cars one. In order to achieve more stable results, we added an additional step, which consisted in trying different pedestrian starting locations (*i.e.*, north, west, south) and averaging the detection outcome of the four configurations. Notice that the use of any compression and pre-processing methods is out of the scope of the paper. Each point cloud is

about 805 kB but spatial and temporal redundancy can be removed in order to reduce the amount of data to be exchanged among cars close to the crossroad. In [26] some methods to remove the redundancy in time and space are proposed and analyzed.

4.2. Point clouds merging

The first vehicle of the environment is the ego car (*i.e.*, vehicle 1). For this experiment, all the point clouds have to be references using the ego coordinate system. During the simulation, after obtaining each LiDAR scan, we translate and rotate the entire point cloud in respect to the ego car location. This is expressed as $p_c^{(1)}$ *i.e.*, the point cloud from vehicle 1. Similarly, this applies to the other available vehicles. The transformed point clouds coming from the ℓ-th available vehicle *i.e.*, $p_c^{(\ell)}$, are then merged in a round-robin fashion with the previous ones, resulting into P_c defined as follows:

$$P_c = \sum_{\ell=1}^{v} p_c^{(\ell)}, \tag{7}$$

where v represents the number of vehicles involved into the LiDAR obstacle detection. It is worth noting that, more importantly, for increasing values of v, the obstacle detection will be more accurate.

At the end of this stage, we collected a total of four point clouds *i.e.*, from v_1 to v_4, for each snapshot. In the next section we will compare the pedestrian detection performances based on these configurations.

4.3. Pedestrian detection and evaluation

In 3D Object Detection, the aim is to predict three dimensional rotated bounding boxes from a point cloud. The output obtained from the point cloud merge state, is thus used as an input to a neural network in charge to detect the four pedestrians. In particular we compared the detection performances obtained using a single point cloud *i.e.*, data from the ego car, with the merged point clouds *i.e.*, combined LiDARs as from Eq. (7). The neural network pipeline is based on CenterPoint [27], which is a two stages 3D Object Detection

network. CenterPoint relies on the same keypoint estimation approach used in CenterNet [28], which consists in predicting a heatmap $\hat{Y} \in [0,1]^{W \times H \times K}$ where W and H are respectively the width and height of the input image and K is the number of classes. Every local maximum of the predicted heatmap corresponds to an object center. For each of them, CenterNet regresses to a 2D bounding box by means of the size map $\hat{S} \in \mathbb{R}^{W \times H \times 2}$. In almost all neural networks for 3D Object Detection, the point cloud is initially quantized into regular bins, which can be Voxels or Pillars, by the VoxelNet [29] or PointPillars [30] backbones, respectively.

After this stage, an encoder is in charge of extracting features, producing a feature map $M \in \mathbb{R}^{W \times L \times F}$, where W and L are width and length, and F is the number of channels. These dimensions depend on the backbone properties such as size or stride. In the same way as CenterNet, the feature map is used to obtain a heatmap where the peaks correspond to the object centers from a 2D bird-eye view. In the last stage of the neural network architecture, all the other 3D properties (*i.e.*, boxes size and orientation) are regressed starting from the object centers. In particular, this stage is made of a shared two-layer MLP (*i.e.*, multilayer perception), which is in charge of (i) obtaining the class confidence score and (ii) box regression prediction. For every detection, the network outputs a confidence score $D \in [0,1]$, which is compared with a threshold value D_{th}. To complete the evaluation, we took advantage of an existing CenterPoint model pre-trained on the NuScenes dataset [31], which is made of $\approx$ 390k LiDAR sweeps distributed in $1,000$ scenes. The NuScenes car equipment consists of 6 cameras, 1 LiDAR, 5 RADAR, GNSS and IMU. The LiDAR sensor is a Velodyne HDL32E, which has analogue specifications of our simulated one. This similarity allowed skipping the training part and assess the performances directly by inference on the simulated LiDAR sweeps.

At this point, we compared the detector output from the ego car versus the merged point clouds, with the ground truth snapshots made in the simulation environment. The evaluation metric for overlaps is 2D BEV (*i.e.*, bird-eye view)

Intersection over Union (or IoU), which is defined as:

$$IoU = \frac{A_O}{A_U}, \tag{8}$$

where A_O and A_U are the area of overlap and union, respectively. Although it is possible to evaluate the system using a 3D IoU [32], for the purpose of this work we considered adequate a simpler method, such as the classic one presented in the previous paragraph. Once an obstacle has been detected, the proposed system sends the warning message to the other cars approaching the crossroad, and provides them the pedestrian position detected by the LiDAR. Nevertheless, concerning the accuracy necessary to provide the considered service (*i.e.*, pedestrian detection and warning transmission), it is required that the approaching car is aware of the lane in which the pedestrian is crossing. In this way, this car can properly adapt its speed based on its planned route. In the event that the vehicle receives the crossing pedestrian's position that is in the lane is going to occupy, it should reduce its speed accordingly in order not to run over the pedestrian, otherwise it can take its motion unchanged.

In order to assess the performances of object detection, we relied on the commonly used metrics Precision $\mathcal{P}$, Recall $\mathcal{R}$ and $\mathcal{F}_1$ Score. These metrics are derived by first classifying the detector outcome in four distinct categories, which are: T_p (true positives), F_p (false positives), T_n (true negatives) and F_n (false negatives). Specifically, true positives are defined as the number of correctly identified bounding boxes. False positives are the detections that correspond to none of the ground truth boxes. False negatives are defined as the count of boxes, which were not detected. Eventually, true negatives are generally ignored in object detection tasks, and more significant in object classification. In our case, this value is always assumed equal to zero. The mathematical formulations of the considered metrics are expressed as follows:

$$\mathcal{P} = \frac{T_p}{T_p + F_p}, \tag{9}$$

and

$$\mathcal{R} = \frac{T_p}{T_p + F_n}. \tag{10}$$

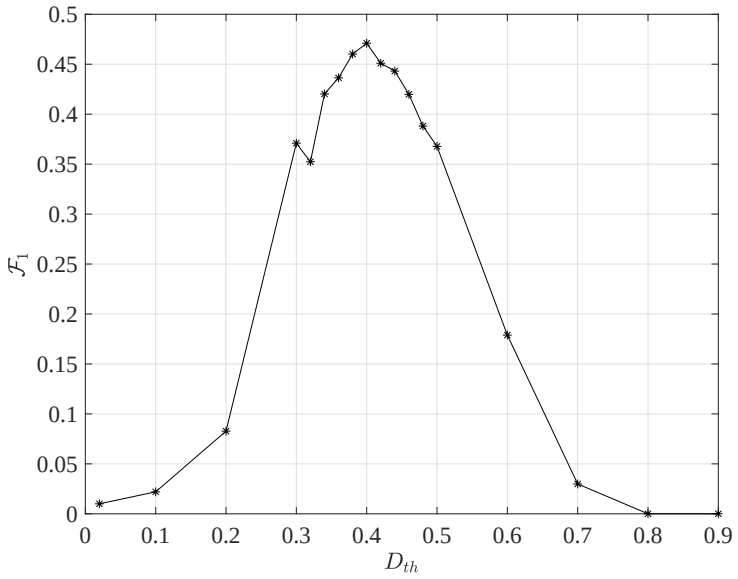


Figure 7: $\mathcal{F}_1$ score versus the detection threshold.

Precision and recall can be combined to define another metric, which is the $\mathcal{F}_1$ Score, defined as the harmonic mean between $\mathcal{P}$ and $\mathcal{R}$. It yields:

$$\mathcal{F}_1 = 2 \cdot \frac{\mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}}, \tag{11}$$

whose highest value is 1, which indicates perfect precision and recall.

5. Simulation results

In this section, the performance assessment of MuSLi will take into account (i) the detection accuracy, and (ii) network KPIs, such as the outage probability and latency related to message forwarding. In the latter case, each of these KPIs has been evaluated for different scenarios and transmission modes.

Regarding the pedestrian detection assessment, we investigated the on-board neural network behavior versus the confidence score threshold for the object detection *i.e.*, D_{th}. As shown by Figure 7, the threshold set to 0.4 maximizes the $\mathcal{F}_1$ score. This value has been considered in the following simulation results.

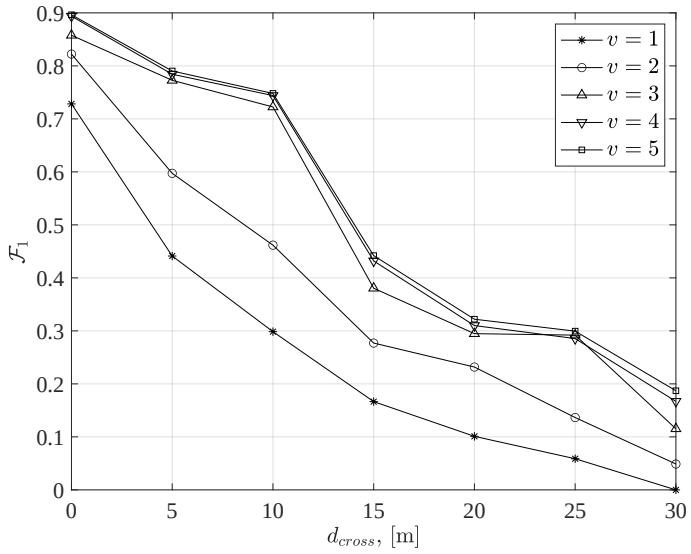


Figure 8: Comparison of the performance results ($\mathcal{F}_1$ score) obtained for a single scan (*i.e.*, $v = 1$) and multiple merged LiDAR scans (*i.e.*, $v \geq 2$), from one and multiple cars, respectively.

We initially evaluate the LiDAR clouds computed in case of (i) traditional single vehicle detection, and (ii) MuSLi approach, with one single and multiple LiDAR sensors mounted on different vehicles, respectively. In Figure 8, a comparison between the single vehicle (*i.e.*, $v = 1$) and multiple vehicles (*i.e.*, $2 \leq v \leq 5$) performances is shown. It is worth noting that for this specific scenario, the performance gain obtained by adding the fourth vehicle is limited, and so we can approximate the most appropriate number of vehicles achieving high performance with MuSLi as 4. Indeed, adding the fifth car in the obstacle detection mechanism does not increase the overall performance gain, and the $\mathcal{F}_1$ score for $v = 5$ is almost the same as the curve for $v = 4$. This result is due to the position of the fifth car, which is laying behind the first car and then its vision is "hindered" by this car. Notice that the choice of $v = 4$ is also due to the need of keeping low the computational cost, as it is linearly dependent on the number of vehicles involved in the object detection. Moreover, the presence of two more cars at the intersection allows the pedestrian detector to reach

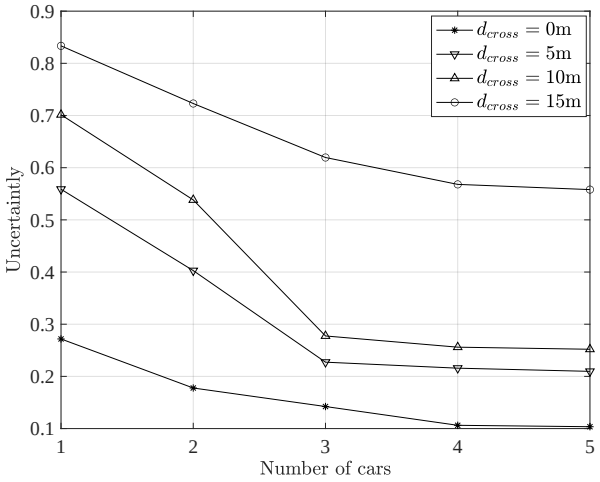


Figure 9: Uncertaintly vs. the number of cars and for different distances to the crossroad.

an $\mathcal{F}_1$ score of 0.72 at the distance of 10 meters. Compared to $\mathcal{F}_1 = 0.29$ obtained from a single vehicle at the same distance, we achieve a performance boost of 0.43, which corresponds to a 148% increase. Instead, if we consider the case which involves only the first two cars, the performance boost is equal to 0.17 that is a 58% increase, which is still enough to consider this kind of approach. Furthermore, from Figure 9 it is evident the uncertainty $\mathcal{U}$ defined as $\mathcal{U} = 1 - \mathcal{F}_1$ is reduced as the number of cars sharing point clouds increases, for example from $\mathcal{U} = 0.28$ to $\mathcal{U} = 0.1$ at $d_{cross} = 0$ m or from $\mathcal{U} = 0.7$ to $\mathcal{U} = 0.25$ at $d_{cross} = 10$ m, corresponding to $v = 1$ and $v = 5$, respectively.

The position accuracy of the LiDAR is quite high (on the order of 20 cm at maximum). In our settings we considered the typical IoU value of 0.5 [27]. Thus, assuming an area of a pedestrian of 50×30 cm^2 (in 2D from high view), the maximum error in localizing a pedestrian is 16.5 cm for $\mathcal{F}_1$ values in Figure 8. Notice that if some phenomena like blind spots occur, we still can obtain a similar accuracy by entirely removing the car from the network elements in charge of the obstacle detection, therefore assuming a total occlusion from that car. It follows that, for partially occluded cars, the $\mathcal{F}_1$ score at a specific distance

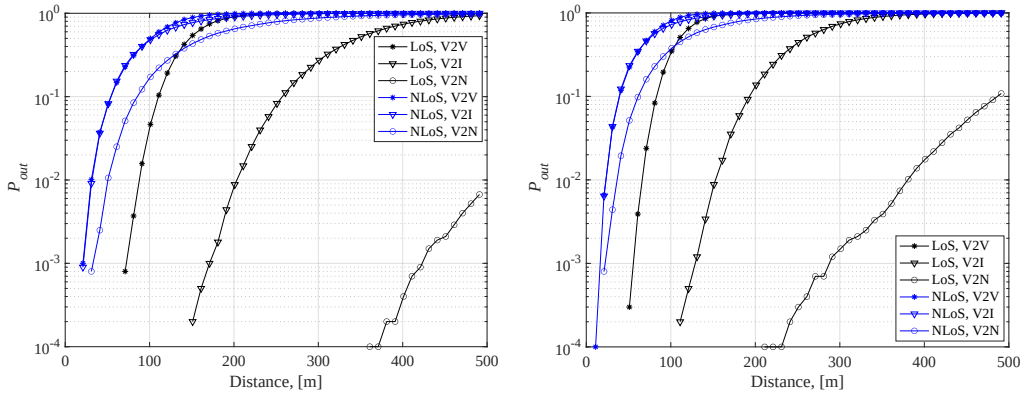
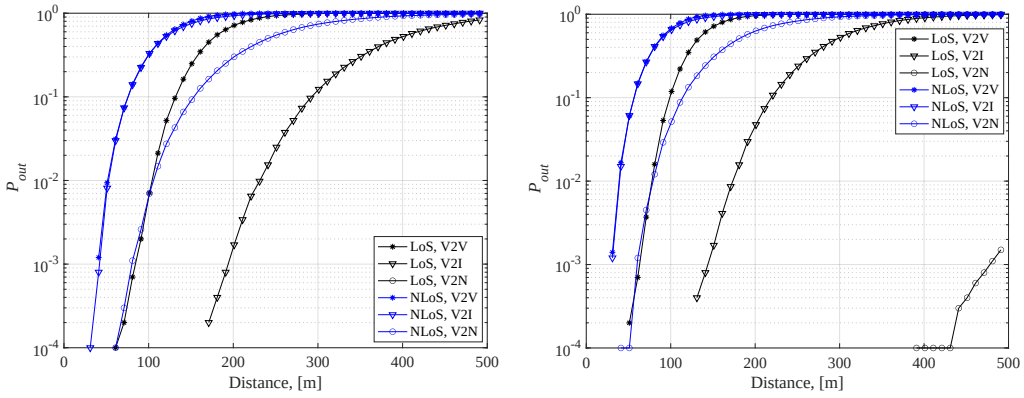


Figure 10: Comparison of the outage probability versus the distance, in case Urban Micro scenario, for (a) low and (b) high interference, respectively.

will fall between the curve computed for v cars *i.e.* $\mathcal{F}_1(v)$, and that for $(v-1)$ cars *i.e.*, $\mathcal{F}_1(v-1)$, where v is the number of the cars participating in the obstacle detection mechanism.

Network performance related to MuSLi technique is described as follows and are based on parameters in Table 1. Figure 10 depicts the outage probability in case of Urban Micro scenario [19] and for LoS and NLoS transmission modes. Also, we assume two different cases for (i) low interference (*i.e.*, $I/\eta + 1 = 5$ dB) and (ii) high interference (*i.e.*, $I/\eta + 1 = 10$ dB), which both refer to noise raise with respect to the thermal noise power, respectively depicted in Figure 10 (a) and (b). Such values represent average lower and upper bounds and have been chosen to take into account the variability of interference due for example to the increase of the number of simultaneous transmitters in the considered area, thus investigating the outage probability ranges.

As expected, we observe that for a given distance, the outage probability increases. For instance, in case of low interference, the outage probability achieved with LoS and V2I communication mode is 10^{-3} at ≈ 170 m, while the same value is reached at shorter distance *i.e.*, ≈ 120 m, in case of high interference.



(a) (b)

Figure 11: Comparison of the outage probability versus the distance, in case of Urban Macro scenario, for (a) low and (b) high interference, respectively.

Furthermore, performance in case of NLoS is worse as compared to LoS, as well as better results are obtained with V2N, followed by V2I, and then V2V. Notice that it depends on the transmission ranges achievable in case of different communication modes. Indeed, in V2V, communications are guaranteed in case of short ranges, reaching an outage probability of 10^{-3} around 70 m, while a similar value is reached with V2N at 420 m, both for LoS and low interference, as shown in Figure 10 (a).

Similar considerations can be taken observing the outage probabilities in case of Urban Macro and Rural scenarios, respectively in Figure 11 and Figure 12. Moving from Urban Micro to Rural, the reachable transmission ranges with outage probability $\leq 10^{-3}$ increase. For instance, in Figure 12 (a) for V2V LoS and Rural scenario at low interference, the outage probability $\leq 10^{-3}$ is guaranteed till 270 m, while the same value of outage probability is guaranteed till 190 m in case of Urban Macro, for V2I LoS and low interference, as depicted in Figure 11 (a). Finally, notice that in same cases, the outage probability for V2N transmission mode is not shown, due to favorable propagation conditions that occur mainly in Rural scenario (see Figure 12).

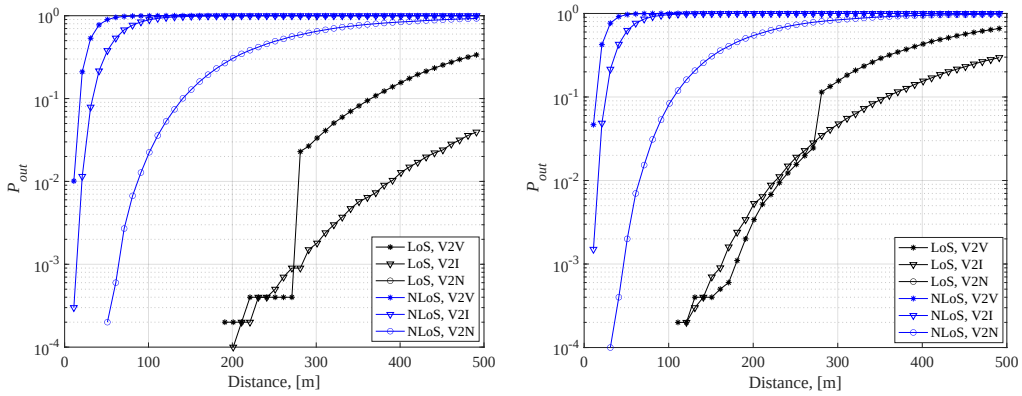


Figure 12: Comparison of the outage probability versus the distance, in case of Rural scenario, for (a) low and (b) high interference, respectively.

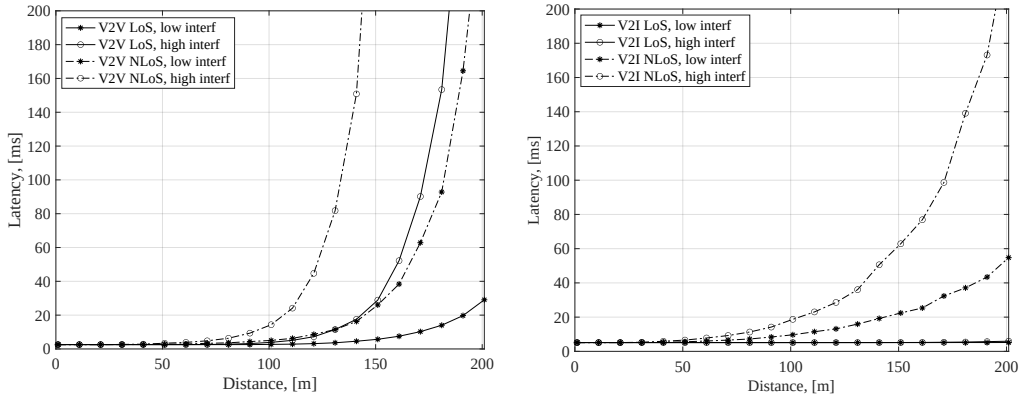
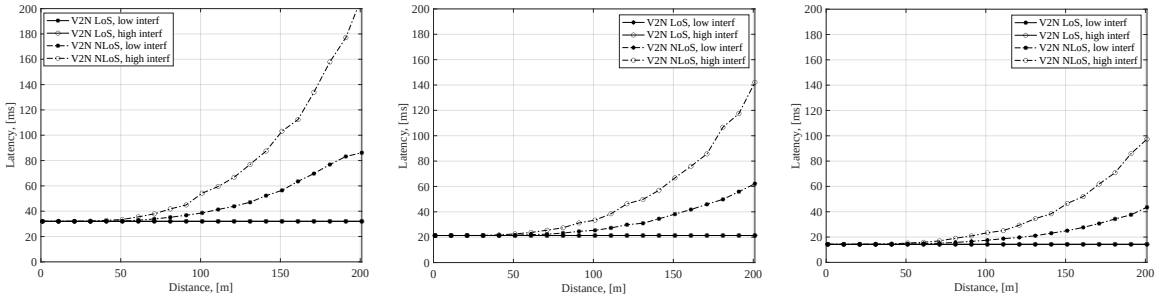


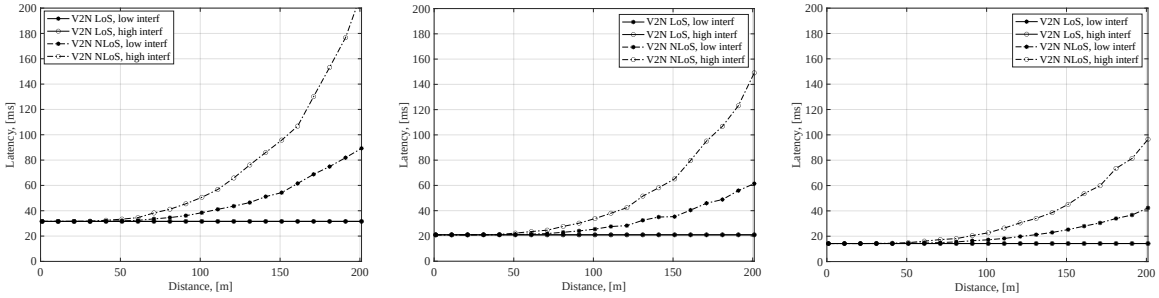
Figure 13: Semi-persistent scheduling. Comparison of the latency vs. the distance from a destination vehicle, for (a) V2V, and (b) V2I connectivity mode.

In the following, we are now describing the performance of message forwarding through MuSLi, in case that an obstacle has been detected. Without lack of generality, we consider only the Urban Micro environment, and the main parameters adopted in our simulations are reported in Table 2. In all the fig-



(a) (b) (c)

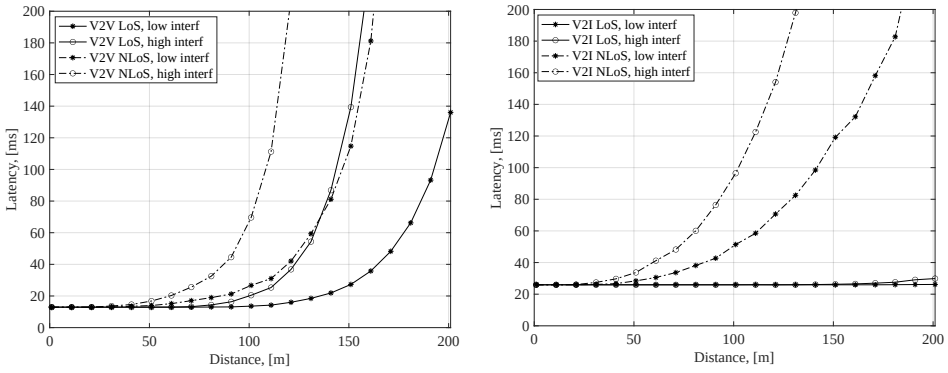
Figure 14: Semi-persistent scheduling. Comparison of the latency vs. the distance from a destination vehicle, in case of V2N connectivity mode via Cloud, for (a) 15kHz, (b) 60kHz, and (c) minislot transmission.



(a) (b) (c)

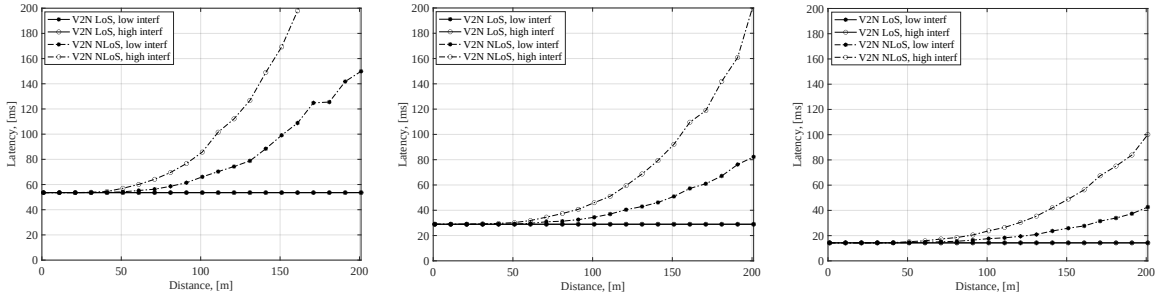
Figure 15: Semi-persistent scheduling. Comparison of the latency vs. the distance from a destination vehicle, in case of V2N connectivity mode via MEC, for (a) 15kHz, (b) 60kHz, and (c) minislot transmission.

ures related to the latency, we considered the 90-th percentile of the related CDFs, as depicted in Figure 4. Similar results and considerations can be obtained in the other environments (*i.e.*, Urban Macro and Rural), not reported for brevity. Notice that the number of retransmissions affects the latency according to the experienced outage probability, as expressed in Eq. (3). We can observe that for the typical operation points of the communication system (*e.g.*, for $P_{out} \leq 0.2$), the number of re-transmissions is very low (*i.e.*, approximately



(a) (b)

Figure 16: Dynamic scheduling. Comparison of the latency vs. the distance from a destination vehicle, for (a) V2V, and (b) V2I connectivity mode.



(a) (b) (c)

Figure 17: Dynamic scheduling. Comparison of the latency vs. the distance from a destination vehicle, in case of V2N connectivity mode via Cloud, for (a) 15kHz, (b) 60kHz, and (c) minislot transmission.

1.25 for $P_{out} = 0.2$), while for $P_{out} = 0.9$ (unrealistic case) the average number of retransmissions is 10. In the following simulations, we set the maximum number of retransmissions to 10.

In Figure 13 we have computed the message transmission latency for each transmission mode. We observe that latencies are slightly constant for low distances, while they increase as retransmissions are required when the outage

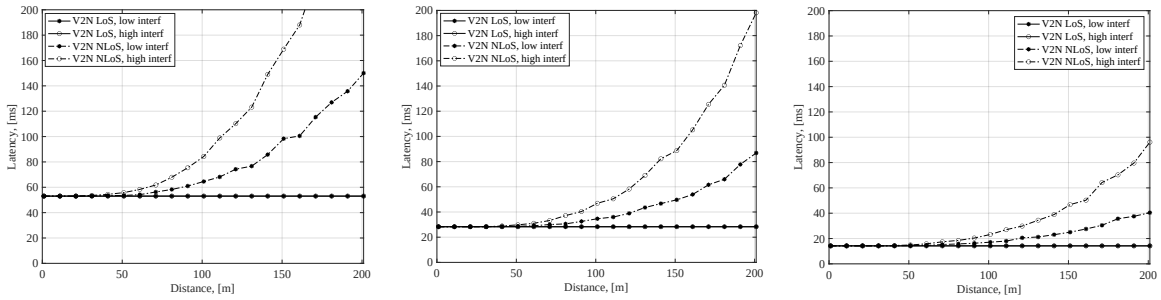


Figure 18: Dynamic scheduling. Comparison of the latency vs. the distance from a destination vehicle, in case of V2N connectivity mode via MEC, for (a) 15kHz, (b) 60kHz, and (c) minislot transmission.

probability is not anymore negligible. The slope (*i.e.*, smooth or fast) and the initial value (*i.e.*, smaller or higher) of latency are both affected by the communication modes, the noise due to interferents, and the scheduling techniques. Specifically, the semi-persistent scheduling provides lower latency than the case of dynamic scheduling due to absence of scheduling request procedure.

From Figure 13 to Figure 15 latencies are for semi-persistent scheduling case. In Figure 13 (a) and (b) latency for V2V and V2I are reported, respectively. In Figure 14 latency in case of V2X application is located in the cloud for (a) 15 kHz, (b) 60 kHz, and (c) minislot transmission respectively, while in Figure 15 latency if the V2X application is located in a closer MEC for (a) 15 kHz, (b) 60 kHz, and (c) minislot transmission respectively.

Considering for example the NLoS and high interference cases, V2V communications mode allows to have lower latency up to $d = 110$ m, guaranteeing a latency lower than 20 ms. V2I overcomes V2V from $d = 110$ m up to $d = 190$ m and for higher distances V2N-MEC [3] at 15 kHz is preferable since retranmissions are lower than those in V2V and V2I. V2N-MEC can experience lower latency

[3]Results of V2N-Cloud are not considered here, since they are slightly higher than those of V2N-MEC.

in case of 60 kHz and minislot, which guarantees a latency between 20 ms and 80 ms in the range from $d = 110$ m and $d = 180$ m, overcoming V2I in the same range.

Similar to the previous organization, from Figure 16 to Figure 18 latencies are for dynamic scheduling case. Latency for V2V in Figure 16 (a), for V2I in Figure 16 (b), for V2N-Cloud in Figure 17 (a) 15 kHz, (b) 60 kHz, and (c) minislot transmission) and for V2N-MEC in Figure 18 (a) 15 kHz, (b) 60 kHz, and (c) minislot transmission) are reported, respectively. Considering also in this case the example of NLoS and high interference values, V2V presents lower latencies (lower than 20 ms) but for reduced distancies (*i.e.*, $d \leq 85$ m). For $d > 85$ m, V2N-MEC at 60 kHz provides lower latencies for the alert message delivery. In case of mini-slot, V2N results comparable to V2V up to 50 m and overcomes it for greater distances.

6. Conclusions

Due to the increasing popularity of connected vehicles, it is becoming essential to report the presence of obstacles on the roadway automatically by using on board advanced sensors such as infrared, electro-optical, acoustic, LiDAR, sonar, radar, etc. In this paper we presented the MuSLi technique, aiming to detect obstacles in a vehicular environment. Vehicles, defined as content islands, subscribe to an extended sensor service when approaching to an area of interest, and publish the contents collected by LiDAR sensors in order to improve the detection accuracy. In case of pedestrian detection, a source vehicle sends alert messages to other approaching vehicles through C-V2X. MuSLi is evaluated in case of specific propagation environments, transmission modes (*i.e.*, LoS and NLoS, the type of implemented scheduling and the amount of interference power in the considering area.

Results are expressed in terms of outage transmissions and latency to deliver the alert messages. For example, they showed that V2V should be preferred up to 115 m in an Urban Micro environment in case a SPS strategy is adopted,

while switching to V2I up to 160 m and to V2N (with SCS of 60 kHz) for greater distances of the approaching vehicle to the considered crossroad. In case of dynamic scheduling V2V should be preferred up to 85 m, switching to V2N (with SCS of 60 kHz) for greater distances.

As future work, we can consider LiDAR sensors mounted on fixed positions along the road, such as on RSUs placed on the traffic lights. In this case RSUs can "observe" the road from a position without obstacles *i.e.*, in LoS. However, a few limitations can occur since the RSUs deployment is not always possible, such as in rural area where the vehicular density is low and pedestrian cannot be forced to cross the road to pedestrian crossings. On the other side, RSUs equipped with LiDAR sensors are useful in scenarios with low visibility due to road bends or bumps. Then, the deployment of LiDAR-based infrastructure nodes should be accordingly designed. Furthermore, the use of unmanned aerial vehicles can be investigated to improve the warning message transmission performance by properly locating them in specific areas.

References

[1] G. Liu, Y. Huang, N. Li, J. Dong, J. Jin, Q. Wang, N. Li, Vision, requirements and network architecture of 6g mobile network beyond 2030, China Communications 17 (9) (2020) 92–104. doi:10.23919/JCC.2020.09.008.

[2] J. Zhu, M. Zhao, S. Zhang, W. Zhou, Exploring the road to 6g: Abc ? foundation for intelligent mobile networks, China Communications 17 (6) (2020) 51–67. doi:10.23919/JCC.2020.06.005.

[3] B. Huang, J. Zhao, J. Liu, A survey of simultaneous localization and mapping with an envision in 6g wireless networks, arXiv: Robotics (2019).

[4] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, Y.-J. A. Zhang, The roadmap to 6g: Ai empowered wireless networks, IEEE Communications Magazine 57 (8) (2019) 84–90. doi:10.1109/MCOM.2019.1900271.

[5] F. Tang, Y. Kawamoto, N. Kato, J. Liu, Future intelligent and secure vehicular network toward 6g: Machine-learning approaches, Proceedings of the IEEE 108 (2) (2020) 292–307. doi:10.1109/JPROC.2019.2954595.

[6] S. V. Maru, V. R. Shah, R. H. Jhaveri, Obstacle detection for vehicles in intelligent transport system, in: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 431–435.

[7] A. Iqbal, Obstacle detection and track detection in autonomous cars, in: S. Ersoy, T. Waqar (Eds.), Autonomous Vehicle and Smart Traffic, IntechOpen, Rijeka, 2020, Ch. 5. doi:10.5772/intechopen.89917.
URL https://doi.org/10.5772/intechopen.89917

[8] A. Mukhtar, L. Xia, T. B. Tang, Vehicle detection techniques for collision avoidance systems: A review, IEEE Transactions on Intelligent Transportation Systems 16 (5) (2015) 2318–2338. doi:10.1109/TITS.2015.2409109.

[9] D. Habermann, C. Garcia, Obstacle detection and tracking using laser 2d, in: 2010 Latin American Robotics Symposium and Intelligent Robotics Meeting, IEEE, 2010, pp. 120–125.

[10] A. K. Aijazi, P. Checchin, L. Trassoudaine, Multi sensorial data fusion for efficient detection and tracking of road obstacles for inter-distance and anti-colision safety management, in: 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), IEEE, 2017, pp. 617–621.

[11] J. Han, D. Kim, M. Lee, M. Sunwoo, Enhanced road boundary and obstacle detection using a downward-looking lidar sensor, IEEE Transactions on Vehicular Technology 61 (3) (2012) 971–985.

[12] S. Patra, P. Manzoni, C. T. Calafate, W. Zamora, J.-C. Cano, Leveraging a publish/subscribe fog system to provide collision warnings in vehicular networks, Sensors 19 (18) (2019). doi:10.3390/s19183852.
URL https://www.mdpi.com/1424-8220/19/18/3852

[13] M. K. Madisa, M. K. Joseph, Android and cloud based traffic control system, in: 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), 2018, pp. 1–4. doi:10.1109/ICABCD.2018.8465443.

[14] S. Taghizadeh, P. Jamborsalamati, M. J. Hossain, J. Lu, Design and implementation of an advanced vehicle-to-vehicle (v2v) power transfer operation using communications, in: 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 2018, pp. 1–6. doi:10.1109/EEEIC.2018.8494480.

[15] A. Bazzi, A. O. Berthet, C. Campolo, B. M. Masini, A. Molinaro, A. Zanella, On the design of sidelink for cellular V2X: a literature review and outlook for future, IEEE Access (2021).

[16] ETSI, 5G; Service requirements for enhanced V2X scenarios (3GPP TS 22.186 version 16.2.0 Release 16) (Nov. 2020).

[17] 3GPP, Application layer support for Vehicle-to-Everything (V2X) services (TS 23.286 Version 17.4.0 Release 17) (Jun. 2022).

[18] P. Manzoni, E. Hernández-Orallo, C. T. Calafate, J.-C. Cano, A proposal for a publish/subscribe, disruption tolerant content island for fog computing, in: Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects, SMARTOBJECTS '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 47?52. doi:10.1145/3127502.3127511.
URL https://doi.org/10.1145/3127502.3127511

[19] Technical Report, ETSI TR 138 901 V14.3.0 - 5G; Study on channel model for frequencies from 0.5 to 100 GHz, 3GPP TR 38.901 version 14.3.0 Release 14 (2018).

[20] M. T. Kawser, N. I. B. Hamid, M. N. Hasan, M. S. Alam, M. M. Rahman, Downlink SNR to CQI Mapping for Different Multiple Antenna Techniques in LTE, International journal of information and electronics engineering 2 (5) (2012) 757.

[21] 3GPP, Physical layer procedures (3GPP TS 36.213 version 16.9.0 Release 16), Sec. 7.2 (Sep. 2022).

[22] H. Holma, A. Toskala, T. Nakamura, 5G technology: 3GPP new radio, John Wiley & Sons, 2020.

[23] R. Giuliano, F. Mazzenga, E. Innocenti, F. Fallucchi, I. Habib, Communication network architectures for driver assistance systems, Sensors 21 (20) (2021) 6867.

[24] M. H. C. Garcia, A. Molina-Galan, M. Boban, J. Gozalvez, B. Coll-Perales, T. Şahin, A. Kousaridas, A tutorial on 5G NR V2X communications, IEEE Communications Surveys & Tutorials (2021).

[25] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: An open urban driving simulator, in: Proceedings of the 1st Annual Conference on Robot Learning, 2017, pp. 1–16.

[26] Q. Wang, L. Jiang, X. Sun, J. Zhao, Z. Deng, S. Yang, An efficient lidar point cloud map coding scheme based on segmentation and frame-inserting network, Sensors 22 (14) (2022) 5108.

[27] T. Yin, X. Zhou, P. Krähenbühl, Center-based 3d object detection and tracking, CoRR abs/2006.11275 (2020). arXiv:2006.11275.
URL https://arxiv.org/abs/2006.11275

[28] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, CoRR abs/1904.07850 (2019). arXiv:1904.07850.
URL http://arxiv.org/abs/1904.07850

[29] Y. Zhou, O. Tuzel, Voxelnet: End-to-end learning for point cloud based 3d object detection, CoRR abs/1711.06396 (2017). arXiv:1711.06396.
URL http://arxiv.org/abs/1711.06396

[30] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, O. Beijbom, Pointpillars: Fast encoders for object detection from point clouds, CoRR abs/1812.05784 (2018). arXiv:1812.05784.
URL http://arxiv.org/abs/1812.05784

[31] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuscenes: A multimodal dataset for autonomous driving, CoRR abs/1903.11027 (2019). arXiv:1903.11027.
URL http://arxiv.org/abs/1903.11027

[32] J. Xu, Y. Ma, S. He, J. Zhu, 3d-giou: 3d generalized intersection over union for object detection in point cloud, Sensors 19 (19) (2019). doi:10.3390/s19194093.
URL https://www.mdpi.com/1424-8220/19/19/4093