

An Edge-based Framework for Enhanced Road Safety of Connected Cars

Original

An Edge-based Framework for Enhanced Road Safety of Connected Cars / Malinverno, M.; Manges-Bafalluy, J.; Casetti, C.; Chiasserini, C. F.; Requena-Esteso, M.; Baranda, J.. - In: IEEE ACCESS. - ISSN 2169-3536. - STAMPA. - 8:1(2020), pp. 58018-58031. [10.1109/ACCESS.2020.2980902]

Availability:

This version is available at: 11583/2801172 since: 2020-04-02T10:32:16Z

Publisher:

IEEE

Published

DOI:10.1109/ACCESS.2020.2980902

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

An Edge-based Framework for Enhanced Road Safety of Connected Cars

MARCO MALINVERNO¹, (Member, IEEE), JOSEP MANGUES-BAFALLUY², CLAUDIO CASETTI¹, (Senior Member, IEEE), CARLA FABIANA CHIASSERINI¹, (Fellow, IEEE), MANUEL REQUENA-ESTESO², and JORGE BARANDA², (Senior Member, IEEE)

¹Politecnico di Torino, Torino, Italy (e-mail: firstname.lastname@polito.it)

²CTTC, Barcelona, Spain (e-mail: firstname.lastname@cttc.cat)

Corresponding author: M. Malinverno (e-mail: marco.malinverno@polito.it).

C. Casetti and C. F. Chiasserini are also with CNIT, and C. F. Chiasserini with CNR-IEIIT. This work was supported in part by the EU Commission through the H2020 5G-TRANSFORMER project (Grant No. 761536), and in part by the CARS Interdepartmental Lab at Politecnico di Torino, by Spanish MINECO grant TEC2017- 88373-R (5G-REFINE) and by Generalitat de Catalunya grant 2017 SGR 1195.

ABSTRACT In this paper, we present an enhanced Collision Avoidance (eCA) service that leverages vehicle connectivity through a cellular network to avoid vehicle collisions and increase road safety at intersections. The eCA service is assumed to be deployed at the edge of the network, thus curbing the latency incurred by the communication process. The core of the eCA service is composed of a Collision Avoidance Algorithm (CAA), and a Collision Avoidance Strategy (CAS). The former predicts the vehicle's future trajectory through the positional information advertised by periodic beacons and detects if two vehicles are on a collision course. The latter decides which of the vehicles potentially involved in a collision should yield. The vehicles are then notified of both the impending danger and of the actions needed to avoid it. We have simulated our solution using SUMO (Simulation of Urban MObility) and ns-3 (network simulator 3) with the LENA (LTE-EPC Network simulAtor) framework on a Manhattan-grid road topology, and observed its good performance in terms of avoided collisions percentage as a function of vehicle speed and different vehicles densities.

INDEX TERMS Advanced driver assistance systems, Automated vehicles, Collision avoidance, Edge computing, Intelligent vehicles, Vehicle safety

I. INTRODUCTION

The number of traffic fatalities on roads remains unacceptably high. With a peak of 1.35 million, as reported in the "Global status report on road safety 2018", traffic accidents represent the 8th leading cause of death for people of all ages and the 1st for children and young adults from 5 to 29 years of age [1]. The severity of this problem, in terms of deaths and injuries, is often under-estimated by governments and local authorities. The under-reporting of traffic accidents fatalities is still common in many regions of the world, and this often results in a low priority being given to road safety with respect to other public health concerns that are less deadly, but have higher media coverage. For these reasons, the new generations of connected vehicles must respond to the increasing request for safer and smarter roads.

The communication capabilities introduced by vehicular networks have paved the way to a new set of applications,

ranging from infotainment to safety, which will completely change our daily transportation experience. While standardization bodies, such as IEEE and 3GPP, are working to improve their protocols for vehicle connectivity, the first Vehicle-to-Everything (V2X)-enabled chips are hitting the market, with several car-makers starting to massively integrate them in their vehicles. An important role in the transition to the V2X paradigm will be played by the cellular network (i.e., Long Term Evolution (LTE) and 5G), which is the main candidate to support vehicular applications, thanks to its widespread infrastructure and the reliability guarantees it provides. Furthermore, using the network edge ensures low-latency communications, which become critical for the support of safety applications and autonomous driving. Actually, the programmability and virtualization capabilities introduced in 5G networks by Software Defined Networking (SDN) and Network Function Virtualization (NFV) will

allow dynamically allocating networking, computing and storage resources, hence satisfying the constraints imposed by different vehicular applications. Indeed, such applications will have to adapt to varying car traffic conditions, network infrastructure conditions, and to their own performance. 5G-TRANSFORMER [2] is a European research project in which dynamic deployment at the edge (e.g., based on location constraints) and adaptation of automotive services (e.g., scaling out [3]) was evaluated by exploiting the ETSI (European Telecommunications Standards Institute) NFV-based platform developed in the project. This also sets the general framework of this paper.

More specifically, in this paper, we address road-safety services and focus in particular on vehicle Collision Avoidance at intersections, in which an application server deployed at the edge of the network infrastructure (i.e., close to vehicles) gathers all the Cooperative Awareness Messages (CAMs) periodically sent by the vehicles, runs a Collision Avoidance application detecting vehicles on a collision course, and transmits Decentralized Environmental Notification Messages (DENMs) to the vehicles possibly involved in the detected collision. In this context, we provide the following main contributions:

- (i) we design an edge-based, enhanced Collision Avoidance (eCA) service, which, thanks to its architecture, allows the investigation and the selection of strategies to avoid vehicle collisions and increase road safety at intersections;
- (ii) we develop an open source simulation framework that closely mimics real-world conditions and dynamics, combining a realistic cellular network model (ns-3 LENA) and realistic mobility traces (SUMO). Importantly, our framework is set to be made publicly available, so that other researchers can reuse the source code and enhance the application, or develop new road-safety applications;
- (iii) we present the results of several simulation campaigns, showing the performance of the eCA service when different road safety strategies are adopted. Interestingly, we demonstrate not only that our service drastically reduces the number of road accidents at crossroad intersections, but also that V2X technologies are the key-enabler for efficient vehicle traffic management, as it greatly increases the average vehicle speed compared to traffic-light regulated intersections.

The rest of the paper is organized as follows. Section II discusses the state of the art on safety applications for connected vehicles as well as the network architecture considerations to take into account, while Section III introduces our system architecture (including the eCA architecture) and Section IV then describes the strategies adopted by the eCA service. As for its evaluation, the following sections present the simulation framework (V), the scenario parameters (VI), and the performance metrics (VII). The results on these metrics obtained through such a framework are presented in

Section VIII. Finally, Section IX concludes the paper.

II. RELATED WORK

The use of computing resources at the edge of the network infrastructure to support automotive services has been investigated in, e.g., [4]–[9]. In particular, [4] proposes a two-level edge computing for automated driving services, where the computational effort is split between the vehicles and the edge server at the base station. The focus of [4], however, is on content delivery, and on how content caching at the edge can improve resource utilization and quality of service. The work in [5], instead, proposes a collaborative vehicular edge computing framework that enables scalable vehicular applications.

In the context of edge computing, an important role has been played by Multi-access Edge Computing (MEC), which provides a reliable computational platform and can effectively exploit radio related-tasks and performance. In [6], Avino et al. developed a MEC platform based on Open Air Interface (OAI) to support a class of safety services, including vehicle collision avoidance. Furthermore, [7] highlights how MEC platforms can be used to support handover prediction, by leveraging the information coming from road users. The shortcomings due to the limited resources available in the MEC are instead highlighted in [9], where a hierarchical cloud-based Vehicular Edge Computing (VEC) is proposed as a solution.

Another body of work underlines the role of cellular networks to support and back up the service provisioning of critical applications, such as vehicle collision avoidance. For instance, [10] investigates the possibility to use LTE for vehicular applications, stressing its weaknesses and strengths, and surveying the current efforts made to address these issues. Undoubtedly, automotive services can benefit from the wide deployment of cellular network infrastructure and from its reliability, while providing very low latency [11]. This has been confirmed by the work done within several European projects, such as 5G-TRANSFORMER [2], which has experimentally demonstrated the dynamic deployment and real-time adaptation (e.g., scaling) to the automotive service requirements and the good performance of the cellular edge in this respect [3].

Additionally, [2], as well as the 5G-CAR project [12], presents a very good overview of the main automotive applications and use cases. Among these, collision avoidance is one of the most prominent, due to the high impact that road fatalities have in our daily life [6], [13]–[16]. The CAA algorithm used in this work, was deployed in [17]–[19], but under different scenarios (e.g., including pedestrians) or in the presence of a distributed architectures. A different CAA algorithm involving vulnerable users is presented in [14], where however the focus is on smartphone power consumption. A different, though related, service is forward collision avoidance, for which [16] proposes a V2V-based solution, leveraging the information exchanged directly between vehicles. In [13], a survey of all the vision-based detection

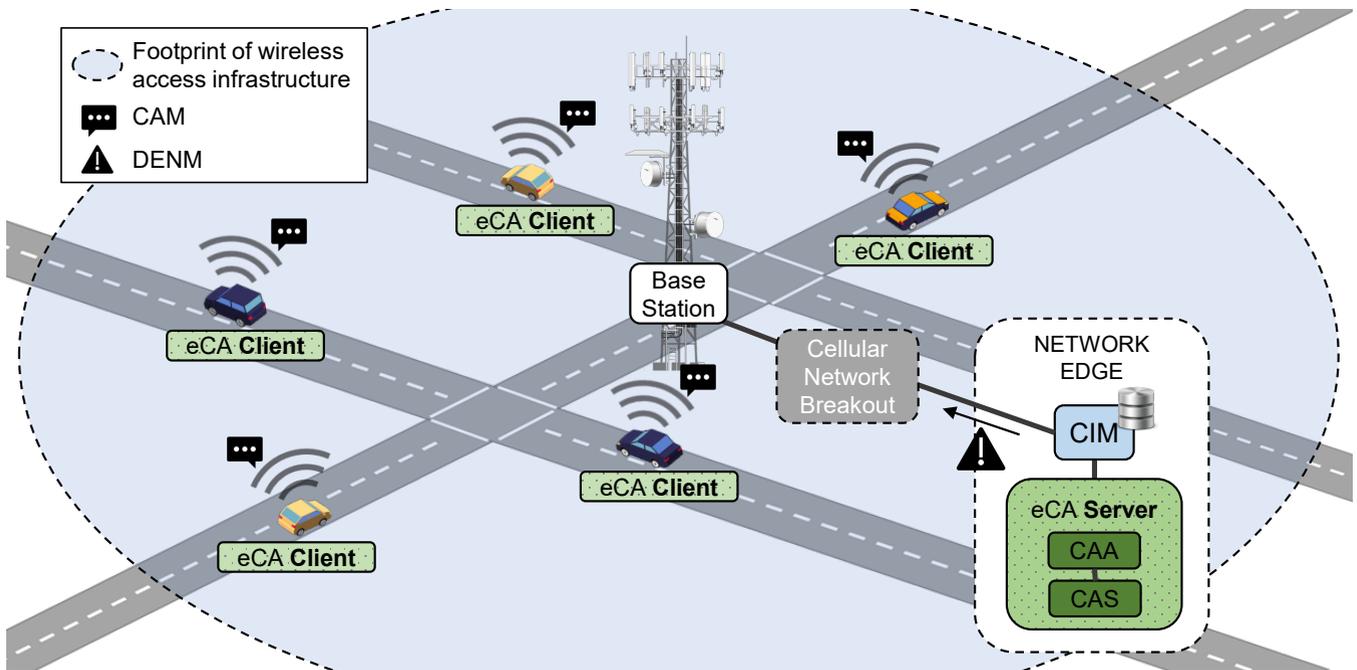


FIGURE 1: The eCA system scenario and service architecture.

techniques for collision avoidance is presented, highlighting how the sensors with which vehicles are equipped, can be exploited to effectively provide this service. Finally, [15] presents and validates, through experimental results, a control structure that integrates path tracking, vehicle stabilization, and collision avoidance – three objectives that sometimes are in conflict.

To the best of our knowledge, none of the existing works has proposed a retro-activated mechanism to avoid collisions at road intersections, leveraging the information carried by CAMs. Most of the collision avoidance applications that carmakers already deploy in their vehicles are based on costly, vision-based technology, such as cameras or lidars. Importantly, these are proprietary solutions whose details are not publicly available, hence a comparison is not possible.

One of the major contributions of this work resides indeed in the CAS logics, which uses the information exposed by the CAA to manage the vehicle right-of-way, thus avoiding collisions.

III. SYSTEM ARCHITECTURE

We consider a road topology with intersections, under the coverage of the eCA system. As depicted in Figure 1, one or more base stations provide connectivity over the geographical area and ensure data communication with passing-by vehicles, which are equipped with On Board Units (OBU). Note that intersections may be unregulated or regulated by traffic lights: in both cases our solution is effective, since, as reported in [20]–[22], it is unfortunately quite frequent that drivers do not obey traffic light signals. Furthermore, as reported in a study conducted by NHTSA (National Highway Traffic Safety Administration), more than 50% of the total

intersection-related crashes occur at intersections that are regulated by traffic lights [23].

The eCA application is designed in a client-server fashion, with the vehicles acting as clients and the server being located at the network edge. Messages coming from the vehicles first reach the base station providing connectivity to the vehicle and then traverse the cellular network to eventually reach the edge node. In a general setup, this happens through the local breakout building block in Figure 1, which allows data to eventually reach the server. As an example, if an ETSI MEC architecture is in place, the local breakout block can correspond to an SDN switch previously configured with the corresponding traffic filtering rules. Such rules let vehicle messages reach the eCA server, which in this case runs as a MEC application in the MEC platform. As a second example, a complete core network can be deployed at a site close to where the vehicles are. In this case, the messages from the vehicles would traverse the core network and reach the edge server through the gateway of the cellular network, without the need for breaking the internal tunnels established in the cellular network.

In any case, vehicles periodically transmit messages, namely, the CAM messages specified by ETSI, carrying the vehicle speed, acceleration, position, heading, and identifier (ID). The CAMs are received at the network edge server where the eCA service is deployed and the vehicle data included therein are extracted and stored in the Cooperative Information Management database (CIM). The CIM may make such information available to a number of applications, including the eCA. Importantly, the logic block composed of the CIM and the eCA application is deployed at the network edge, ensuring low-latency and high-reliability.

The core of the eCA service is composed of the Collision Avoidance Algorithm (CAA) and the Collision Avoidance Strategy (CAS). The former receives as input the vehicle information from the CIM and detects pairs of vehicles that are on a collision course. The CAS, instead, exploits the CAA output to apply decisions on the actions to be taken by the colliding vehicles. In a nutshell, for each pair of potentially colliding vehicles, it determines which of the two should yield, depending on the adopted CAS, and it generates and transmits DENMs accordingly. The DENMs are intended for the vehicles involved in the detected, possible, collision and contain all the information needed to the vehicles to avoid the collision, e.g., type of hazard, detection time, predicted point of collision. Upon receiving a DENM, the vehicle onboard logic present in the eCA client application, warns the driver, or triggers the actuator in case of automated vehicles, which eventually starts the braking phase and stops the car.

IV. THE ENHANCED COLLISION AVOIDANCE SERVICE (ECA)

An overview of the eCA service architecture is shown in Figure 2. The eCA draws on the Collision Avoidance Algorithm (CAA) introduced in [17]–[19]. The general idea of the CAA is to exploit the information contained in CAMs, such as position, velocity, acceleration, and heading, to predict the future trajectory of the vehicle. The information received from the vehicles is stored in the CIM, and then used to build the vehicle trajectories.

Each time the service receives a CAM from a vehicle, it generates the corresponding trajectory, and checks for possible collisions with the other vehicles by using each vehicle's last received information. Note that, since the CAM generation is not synchronous across vehicles, the probability that two or more new CAMs arrive and have to be processed at the same time tends to zero. Also, CAMs are processed in the order in which they are received at the server.

The trajectories are generated by projecting the future positions of the vehicles onto straight or curve segments. In particular, this projection depends on the status of the blinking lights of an vehicle approaching the intersection — an information encoded in CAMs. If the signals of both blinking lights are off, the algorithm projects the future position of the vehicle using a straight line. Otherwise, the trajectory is drawn over multiple segments: over a straight line before the intersection, over a curved line within the intersection, and again over a straight line after the intersection. This turned out to provide a very accurate estimate of the actual trajectory.

Each generated trajectory is analysed and compared with the others, in order to determine if any pair of vehicles is set on a collision course. To this end, two main parameters are considered:

- 1) Space-to-collision (**S2C**), i.e., the minimum distance that will be reached between the vehicles under test, considering that the vehicles follow the projected trajectory;

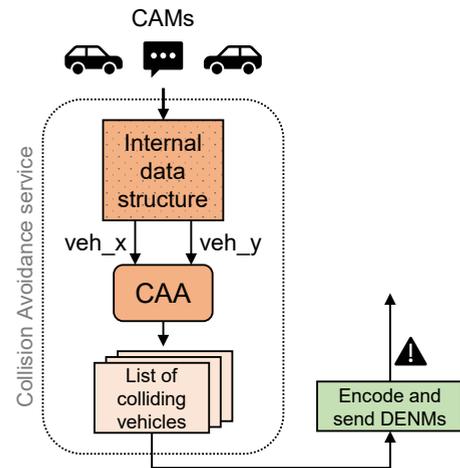


FIGURE 2: General architecture of the eCA service.

- 2) Time-to-collision (**T2C**), i.e., the time that it takes to reach the above S2C between the vehicles under test.

The system detects a possible collision between two vehicles when both S2C and T2C parameters are below some predefined thresholds. The value of such thresholds are dynamically calculated: the S2C threshold depends on the size of the involved vehicles, while the T2C threshold depends on their speed. Once all vehicle pairs have been analyzed, the algorithm returns the list of pairs detected as on collision course.

If one or more of such pairs are detected, the underlying system generates the needed DENMs, each including the type of hazard, the detection time, and the point of collision. DENMs are then forwarded to the messages to the vehicles involved in the detected collision.

The approach taken in the previous works consisted in running the service by feeding the CAA with the CAMs coming from the vehicles, and then in analysing the CAMs received in order to determine whether or not the vehicles could be stopped in time to avoid the collision. In this work, some important functionalities are designed and, accordingly, an additional module is introduced, which further processes the vehicles detected to be on collision course by the CAA. The newly introduced module, called Collision Avoidance Strategy (CAS), identifies which of the vehicles involved in a collision should yield. This decision is then encoded in a `precedence` field inside the DENM sent to the vehicles. This bit is used to make the vehicle stop (`precedence=0`), or to allow it to cross the intersection (`precedence=1`). Upon decoding the message, vehicles will take actions accordingly, i.e., in the eCA service, the vehicles actively react to DENMs by stopping whenever needed.

We propose and investigate two possible implementations of the CAS. One, referred to as Single-Event CAS (SE-CAS), makes decisions only based on each single pair of vehicles that are detected to be on a collision course. The other, called Multiple-Event CAS (ME-CAS), leverages the information

returned by the CAA to build a contention table that will be used to manage the right-of-way of vehicles.

A. SINGLE-EVENT COLLISION AVOIDANCE STRATEGY (SE-CAS)

In SE-CAS, once the CAA returns the list of pairs of vehicles on collision course, each pair is separately analysed and, depending on the configuration adopted, one of these four actions is taken:

- 1) both vehicles are stopped;
- 2) only the vehicle coming from left is stopped;
- 3) only the slowest vehicle is stopped;
- 4) only the farthest vehicle from the crossing is stopped.

Note that the choice to consider several actions reflects the fact that, in real-world situations, different possible actions may be adopted to avoid a collision.

The first configuration, labeled in Figure 3 as “action 1”, is the simplest one: once a collision is detected, each vehicle involved therein is warned by a DENM with the `precedence` field set to 0, so that both vehicles will eventually stop. In the second configuration (“action 2”), the right of way is given to the vehicle coming from the right. The third configuration (“action 3”) stops the slowest vehicle, (i.e., the one that, presumably, can stop in the shortest time), while the last configuration (“action 4”) stops the farthest vehicle from the crossing, i.e., the one that has more space to brake. The whole procedure is detailed in Algorithm 1 (run for each vehicle pair output by the CAA). In the current design of the eCA application, the possible actions have to be predefined and cannot be changed dynamically. However, one of the benefits of our framework resides in its flexibility, and further investigation on the effects of the different actions can easily be integrated inside adaptive strategies, in which

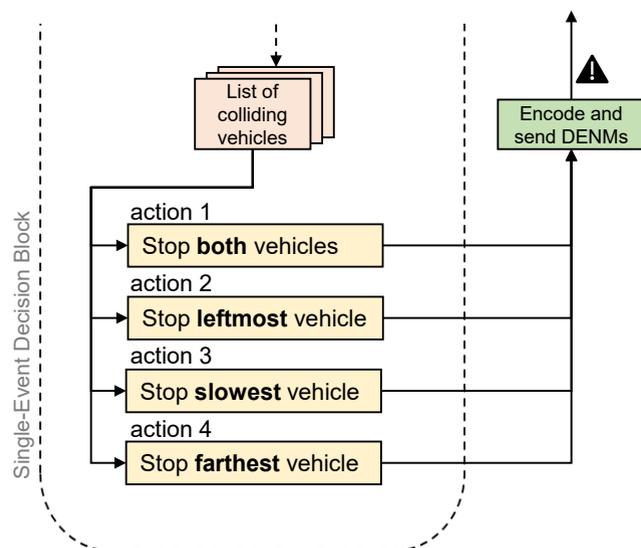


FIGURE 3: The Single-Event Collision Avoidance Strategy (SE-CAS) block.

Algorithm 1 The SE-CAS strategy

DENM₀ => DENM with `precedence` set to 0

DENM₁ => DENM with `precedence` set to 1

Require: $CV, action \triangleright$ Pair of vehicles on collision course, action to be taken

- 1: **if** $action = 1$ **then**
- 2: Send DENM₀ to both vehicles
- 3: **else if** $action = 2$ **then**
- 4: Send DENM₀ to vehicle coming from left
- 5: Send DENM₁ to vehicle coming from right
- 6: **else if** $action = 3$ **then**
- 7: Send DENM₀ to slowest vehicle
- 8: Send DENM₁ to fastest vehicle
- 9: **else if** $action = 4$ **then**
- 10: Send DENM₀ to farthest vehicle
- 11: Send DENM₁ to nearest vehicle

actions are triggered at run-time, according to the vehicle mobility patterns and dynamics.

Independently from the adopted configuration, whenever a vehicle receives a DENM with the `precedence` field set to 0, it starts a braking phase. The braking phase always lasts for a time equal to the *stopping time* of the vehicle. Such time interval is dynamically calculated and is equal to the ratio between the current speed and maximum deceleration. Once the braking phase terminates, the vehicle starts accelerating until it reaches its maximum speed.

Note that SE-CAS suffers from possible deadlocks, especially when the vehicle density is very high. Indeed, in these situations the system halts all the vehicles approaching an intersection, reducing the vehicle traffic efficiency. This may happen, for example, with four vehicles simultaneously approaching a crossing with four arms, each one coming from a different arm: if the system is configured to stop the leftmost vehicle, all the vehicles will be stopped. For this reason, below we introduce a CAS that, although more complex, has a wider perception of the intersection status.

B. MULTIPLE-EVENT COLLISION AVOIDANCE STRATEGY (ME-CAS)

In this case, the list of colliding vehicles returned by the CAA is used to build a contention table, as shown in Figure 4. A table entry is created when the CAA detects a pair of vehicles on collision course. Each entry in the contention table includes the following attributes:

- *Contention Session ID*, a number uniquely identifying a Contention Session (CS);
- *List of vehicles*, list of the IDs of the vehicles involved in the CS;
- *Status*, indicating the status of each vehicle in the CS; the possible values are: running, if the vehicle is crossing the intersection, or stopped, if the vehicle is yielding.
- *Order*, a number representing the order in which the vehicles belonging to a CS will be given right to cross the

intersection (i.e., their status will change from stopped to running).

In general, when a new CS is created, the *Status* of the vehicles on collision course is determined as follows: the farthest vehicle from the center of the intersection yields, thus it is tagged as stopped, while the other is tagged as running. Both vehicles will receive a DENM, but with the precedence field differently set (i.e., 0 if the vehicle has to yield, and 1 otherwise). However, different cases must be addressed:

- 1) if none of the two vehicles that must be inserted belong to an existing CS (Line 2), a new entry is created following the usual procedure, and a DENM is sent to both the vehicles;
- 2) if both vehicles are already in one or more CS (Line 6), a new CS is created only if the two vehicles do not already belong to a same, existing CS (Line 10). In this case, two DENMs should be sent only when both vehicles are tagged as running. In all other cases, no DENM should be sent, since both vehicles are already under the system control;
- 3) if only one of the two vehicles appears already in the table (Line 16), two sub-cases should be considered:
 - a) if the *Status* of the vehicle already in the table is equal to running, then the new vehicle is added to the same CS and the farthest between the two is stopped (Line 19);
 - b) if the vehicle already present in the table is tagged as stopped, then a new CS is created, with the new vehicle tagged as running, and a DENM is sent only to the new vehicle with the precedence field set to 1 (Line 23).

Importantly, as soon as a CAM is received from a vehicle tagged as running, from which it can be inferred that the ego-

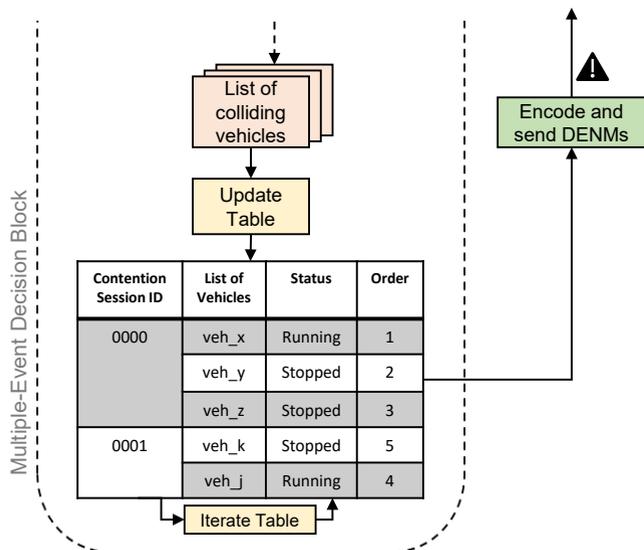


FIGURE 4: The Multiple-Event Collision Avoidance Strategy (ME-CAS) block.

Algorithm 2 The ME-CAS strategy: table update and iteration

```

DENM0 => DENM with precedence set to 0
DENM1 => DENM with precedence set to 1
1: procedure ME-CAS TABLE UPDATE
Require: CV                                     ▷ Pair of vehicles on collision course
2:   if both vehicles of CV not in Table then
3:     Create new CS
4:     Send DENM0 to Vehfar & set Vehfar's status = Stopped
5:     Send DENM1 to Vehclose & set Vehclose's status = Running
6:   else if both vehicles of CV in Table then
7:     if vehicles in same CS then
8:       Return
9:     else
10:      Create new CS
11:      if both vehicles' status = Running then
12:        Send DENM0 to Vehfar & set Vehfar's status = Stopped
13:        Send DENM1 to Vehclose & set Vehclose's status = Running
14:      else
15:        Return
16:      else if one vehicle of CV in Table then
17:        Vehp ← CV[present]
18:        Vehn ← CV[not_present]
19:        if Vehp's status = Running then
20:          Add Vehn to Vehp's CS
21:          Send DENM0 to Vehfar & set Vehfar's status = Stopped
22:          Send DENM1 to Vehclose & set Vehclose's status = Running
23:        else if Vehp is stopped then
24:          Create new CS
25:          Send DENM1 to Vehn & set Vehn's status = Running
26:
27: procedure ME-CAS TABLE ITERATION
Require: Vehexited                               ▷ Vehicle that exited the intersection
28:   for CS | Vehexited in CS do
29:     Remove Vehexited
30:     Add CS to CScheck
31:   for CS in CScheck do
32:     if CS ≠ ∅ then
33:       for Veh in CS do
34:         Vehnext ← vehicle next in order
35:         for CS | Vehnext in CS do
36:           if ∃ Veh in CS with status = Running then
37:             Set Vehnext's status = Stopped
38:             Break
39:           else
40:             Set Vehnext's status = Running
41:             Continue
42:           if Vehnext's status = Running then
43:             Send DENM1 to Vehnext

```

vehicle left the intersection, the procedure to restart another vehicle begins (Line 28).

The exited vehicle is removed from all the CSs it belongs to (Line 29), and all the updated CSs go under a check procedure (Line 31). Among the vehicles that do not have a running companion in any CS, the algorithm selects the one that has been stopped for the longest time (Lines 34–41). Once a vehicle passes this check, it is notified with a DENM with precedence field set to 1 and, finally, restarts (Line 43).

Thus, the order in which the waiting vehicles are processed depends on when they were first inserted in the table, so that the final waiting time is minimized.

At last, we underline that DENMs are used in a different manner in the SE-CAS and in the ME-CAS, thus vehicles need to be aware of which scheme is adopted in order to properly react to DENMs. In particular, under the ME-CAS, DENMs are used to stop and restart the vehicles, similarly to a Virtual Traffic Lights service.

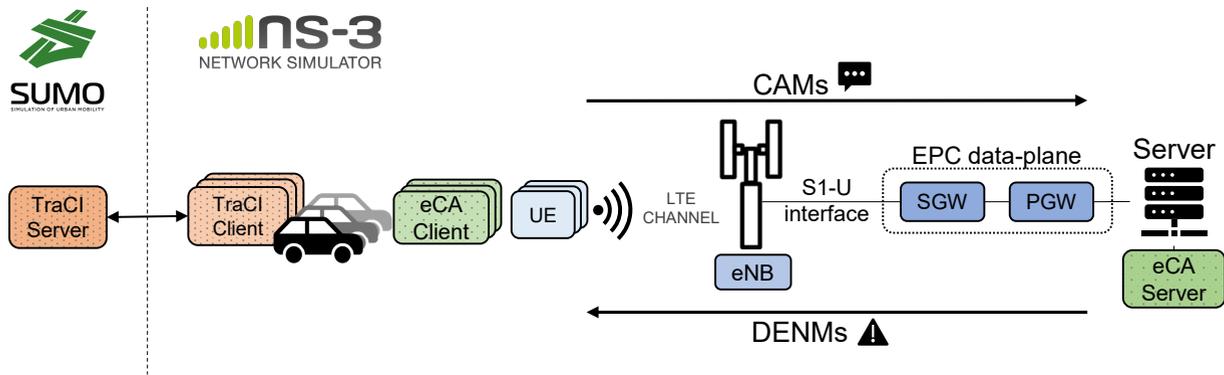


FIGURE 5: High-level architecture of the simulation framework (data plane).

V. SIMULATION FRAMEWORK: DESIGN AND IMPLEMENTATION

The overall high-level architecture of the simulation environment is depicted in Figure 5 (note that, though not depicted, the control plane is also modeled).

When dealing with connected cars, two essential system components need to be modeled:

- A network simulator that models all the aspects of the communication among the various entities and encloses all the layers of the communication stack, from the application to the physical channel. To do so, we leverage ns-3 v3.29 [24], an open-source discrete-event simulator, along with the LENA framework [25], modeling the entire LTE stack.
- A vehicular traffic simulator that manages the node position according to realistic mobility patterns and dynamics. To this purpose, SUMO v1.3.1 [26] is used, which ensures an easy interoperability with ns-3 (and, in general, with several other network simulators) through the TraCI interface [27].

More specifically, the C-V2I communication is provided by leveraging the LTE infrastructure and the OBU aboard the vehicles, representing the LTE user equipment (UEs). The eNB, providing radio access, is connected to the EPC data-plane through the S1-U interface. In its data plane, the EPC is composed of a Serving Gateway (SGW) and a Packet Data Network Gateway (PGW). The PGW gives connectivity to the node running the eCA server, which gathers the CAMs coming from the vehicles and generate unicast DENMs when needed. Thus, the resulting model enables the simulation of an end-to-end IP connectivity through LTE-EPC. In this case, the EPC is deployed at the edge of the network. From the application point of view, each ns-3 node representing a vehicle comes with two modules:

- 1) A TraCI client, which manages the node mobility by retrieving the vehicle position, speed, acceleration, and heading, and that actively takes control of the vehicles by stopping and resuming them when needed. As an example, whenever an ns-3 node representing a vehicle receives a DENM in which the `precedence` field is

TABLE 1: Mobility parameters

Parameter	Value	Description
Vehicle density	[2, 3, 4] veh/km	Number of vehicles per kilometer
Vehicle generation distribution	Poisson with $\lambda = 0.7$	Distribution for vehicles generation
Vehicle dimensions	W1.8 - L4.3	Vehicle width and length
Vehicle speed	[8.33, 13.89, 20.83, 27.78, 36.11] m/s	Vehicle maximum speed
Vehicle acceleration	4 m/s ²	Vehicle maximum acceleration
Vehicle deceleration	7.5 m/s ²	Vehicle maximum deceleration
Map size	3 km	Total length of roads on map
Simulation step length	0.01 s	Mobility updates granularity

set to 0, ns3 communicates with SUMO through the TraCI API, so as to set the vehicle speed to 0. In this way, SUMO forces the vehicle to decelerate, until it completely stops.

- 2) An eCA client, which manages the CAM dissemination and runs the eCA logic upon reception of a DENM.

Importantly, the simulation framework is very flexible and can be used to test any kind of service, with very few modifications.

VI. SIMULATION SCENARIO AND PARAMETERS

The road map used in SUMO is composed of two vertical roads, each 400 meter long, crossed by a central horizontal road, 700 meter long, over which vehicles travel in both directions, for a total of 3-km length. As a result, the topology exhibits six entry lanes and two intersections. The vehicle spatial density and maximum speed are varying parameters in our scenario, while other parameters, such as the maximum acceleration and deceleration, are fixed. The setting of the main mobility-related parameters is reported in Table 1.

All the vehicles enter from one of the six entry lanes, with a generation time following a Poisson distribution with rate $\lambda = 0.7$. As soon as the vehicles leave the scenario, they are randomly rerouted to one of the six entry lanes. The five values of speed we tested are reported in Table 1 in

TABLE 2: Communication network parameters

Parameter	Value	Description
Application message format (CAM and DENM)	ASN.1	ASN.1 decoder and encoder implemented
CAM frequency	10 Hz	Frequency at which CAMs are sent
CAM size	83 B	Dimension of packets at physical layer
DENM size	83 B	Dimension of packets at physical layer
Reaction time	[0.05, 1] s	Delay between the DENM reception and the start of the reaction
Transport-Network layer	UDP-IP	UDP used over IP to create the communication sockets
Fading Model	Trace-based Fading Model	Model used for fading calculation
UE transmission power	23 dBm	Transmission power for UEs
eNB transmission power	45 dBm	Transmission power for eNB
Bandwidth	5 MHz (25 RBs)	LTE channel bandwidth

meters per second, and correspond to 30, 50, 75, 100, and 130 kilometers per hour, respectively; such values represent the vehicle speed in urban, suburban and freeway scenarios. In spite of the urban street layout, we stress the eCA service by testing it with vehicles traveling at high speed. The duration of the simulation step in SUMO is the frequency at which the simulator updates the position of the simulated vehicles. In our case, we set it to 0.01 s, which allows us to have a maximum positioning error, due to the discrete-events nature of the simulator, ranging from 0.08 m (for vehicles running at 8.33 m/s) to 0.36 m (for vehicles running at 36.11 m/s).

The vehicle motion patterns are managed by the SUMO car-following model. The basic idea behind a car-following model is that each vehicle speed depends (i) on the speed of the leading vehicle, (ii) on the space gap between the leading and the following vehicles, and (iii) on other static parameters such as the reaction time or the deceleration. In particular, we adopted the Krauss car-following model, which is the default one in SUMO.

As mentioned, the eCA service has been developed in ns-3 over the LENA framework [25]. The main parameters that have been modified with respect to their default values are reported in Table 2.

CAMs are transmitted at a frequency of 10 Hz, i.e., the maximum frequency allowed by ETSI standards. CAM and DENM messages are encoded using the ASN.1 standard for Intelligent Transportation Systems (ITS), standardized by ETSI [28], [29], which required the implementation of an ASN.1 encoder and decoder in ns-3. The size of CAMs and DENMs is of 83 bytes. They may be bigger than that (up to 800 bytes for CAMs [30]), but we encoded only the information needed to run our service.

We define the reaction time as the interval between the DENM reception at application layer and the starting of the

deceleration phase. The values adopted are:

- 1) 1 s, in case we consider a human driver. Note that the modeling of human reaction time in vehicular scenarios is the subject of extensive research [31]–[33]. Due to the high number of variables involved, such as driver’s mental and physical status, type of warning issued (e.g., haptic, visual, or both), condition of the road, etc., we fixed it to a value obtained averaging over multiple situations;
- 2) 0.05 s, in case we consider a vehicle with an automatic braking system. Specifically, such a value accounts for the interval from the time instant at which the DENM is received at the application layer, to the instant when the vehicle’s actuator starts the deceleration, and it is in accordance with the relevant literature [34], [35].

For what concerns the transport and the network layers of the protocol stack, the communication between eCA clients and the eCA server happens through UDP sockets, which transmit packets over the IP network.

Finally, we introduced a fading model in our simulation framework: to reduce the computational complexity, the fading calculation during simulation run-time is based on pre-calculated traces built using a script inside the LENA framework. Such script takes as input the following main parameters:

- 1) vehicle speed, which affects the Doppler frequency, hence the time-variance property of the fading;
- 2) number of fading paths considered, which characterizes the fading behavior in the frequency domain;
- 3) number of users, which determines the number of fading traces to be generated.

VII. METRICS OF INTEREST

The target of this work is to evaluate our eCA system under different scenarios, in which we varied the speed of the vehicles, their spatial density, the reaction time associated to a DENM reception, and the reaction strategy (with SE-CAS, or with ME-CAS). Unlike previous works [17]–[19], which evaluated the service by only analysing the DENMs received at the vehicles, we perform a study on the possible actions to be taken upon the reception of a DENM, and on how different approaches may impact the system performance, from both the network and the application point of view.

A. NETWORK-RELATED METRICS

To study the network performance of the eCA system, we consider the following metrics:

- **Traffic load**, defined as the network traffic generated by the eCA service, calculated in kb/s. The measurements are made both in the uplink (CAM messages) and in downlink (DENM messages);
- **Latency**, defined as the time gap between the moment at which a CAM is sent, to the moment at which the corresponding DENM is received. This value is computed thanks to the timestamp included in each CAM

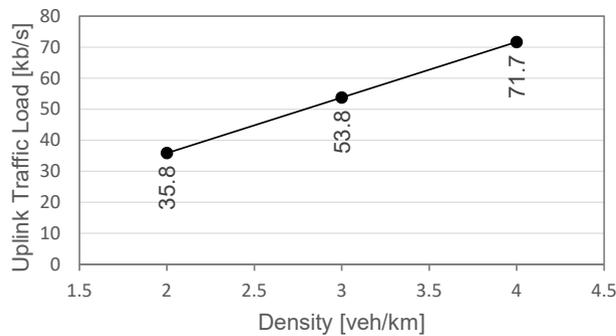


FIGURE 6: Uplink traffic load as a function of the vehicles spatial density.

and DENM, which allow us to discriminate the delay introduced in the uplink scheduling procedure, from the one introduced in downlink.

- **Packet Delivery Ratio (PDR)**, defined as the ratio between the number of packets sent and received by the application, in downlink and in uplink. Given the network architecture deployed, in which the HARQ system ensure re-transmissions, the absence of obstacles, the low number of UEs connected, and the size of the map, the only factor that may impact this metric is fading, which may cause packet losses.

B. APPLICATION-RELATED METRICS

The application performance is studied through the following metrics:

- **Collisions avoided**, defined as the percentage of number of collisions avoided when the eCA service is active, compared to the number of collisions that occur (with the same mobility trace) without the eCA service;
- **Speed**, defined as the average vehicle speed; in the evaluation section, the speed when the eCA service is active is compared to the situation in which two traffic lights are deployed to regulate the traffic at the intersections.

VIII. SIMULATION RESULTS

Here we present the eCA performance, in terms of both network-related and application-related metrics. The results have been obtained with 50 simulation runs, each 300 seconds long, for each combination of speed, density, reaction time, and DENM reaction strategy.

Traffic load. The traffic generated by the application and transmitted on the uplink channel is completely independent of speed, reaction time, and DENM reaction strategy. This is due to the fact that the CAM dissemination rate is fixed at 10 Hz, no matter the speed or the underlying computational logics. Therefore, the results are affected only by the increase of the vehicle spatial density. As shown in Figure 6, the uplink traffic load linearly increases from 35.8 kb/s for 2 vehicles per kilometer, to 71.7 kb/s when the density is doubled.

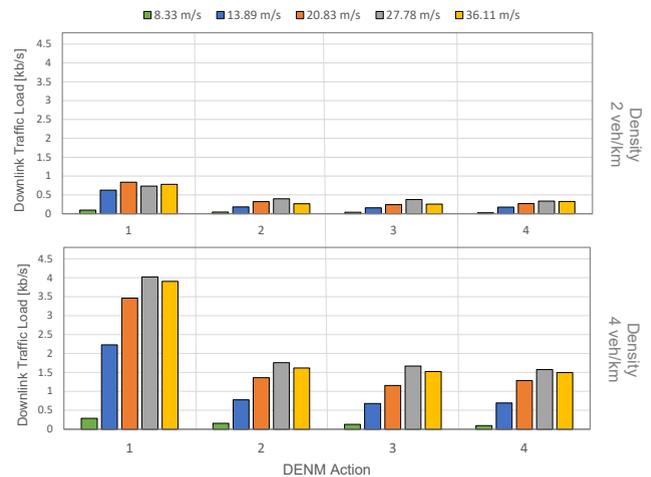


FIGURE 7: Downlink traffic load with SE-CAS. In the x axis the different DENM actions are plotted, with each bar corresponding to a different speed. The two plots correspond to different vehicle spatial densities.

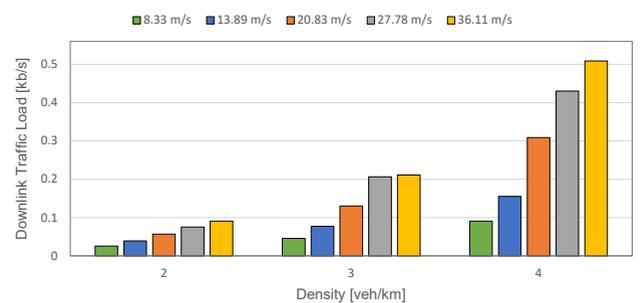


FIGURE 8: Downlink traffic load with ME-CAS as a function of the vehicle spatial density. Each bar corresponds to a different speed.

Looking instead at the downlink traffic, we can see how different strategies result in different traffic loads. This is due to the different number of DENMs needed to instruct vehicles under the different reaction strategies. In general, the application traffic load in downlink is one, or even two, orders of magnitude smaller than in uplink, since DENMs are generated only upon the occurrence of an event.

Figure 7 illustrates the results when deploying the SE-CAS. The x axis represents the different DENM actions (namely, stop both cars, stop the one coming from left, stop the slowest, and stop the farthest from the crossing), while on the y axis we plot the downlink traffic load. Each bar represents a different speed, from 8.33 m/s (i.e., 30 km/h) to 36.11 m/s (i.e., 130 km/h). The two sets of results represent different vehicular spatial densities.

From the plot, we can see how the vehicle spatial density has a big impact on the downlink traffic. To wit, the higher the number of vehicles, the higher the possibilities that dangerous situations occur and, consequently, that the eCA service reacts with a DENM. The relationship is non-linear, indeed by doubling the density, the traffic load increases more than twice (e.g., up to a factor of 7). This is due to the much

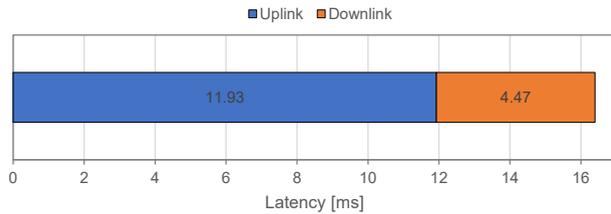


FIGURE 9: Latency experienced in uplink and downlink. The measurements have been made at application layer.

higher number of impending collision situations, since there are many more pairs of potentially colliding vehicles.

By looking at the evolution of the results while changing the DENM action, it is possible to note how the case in which both vehicles are stopped (namely, DENM action equal to 1), is associated to a higher downlink traffic load, thus, to a higher number of DENMs sent. This happens because when both vehicles are stopped, they will occupy the intersection area longer, thus increasing the probability that more pairs of vehicles are detected as on collision course.

If we focus on the evolution of the downlink traffic load when the vehicle maximum speed is increased, we can see that, up to the point in which the vehicle maximum speed is set to 27.78 m/s, we have an expected trend. When the speed is further increased, the value of downlink traffic load tends to decrease, especially with higher vehicle spatial densities. That translates into a lower number of dangerous situation detected by the eCA service, and is due both to the higher positioning error experienced at higher speeds, and to the fact that vehicles occupy the intersection for a shorter time, hence the lower probability that a collision alert is raised.

In Figure 8, the downlink traffic load in the case in which ME-CAS is deployed is plotted. In this case, the x axis reports the increasing vehicle spatial density, while the y axis shows the traffic load in kb/s. It is worth mentioning that using ME-CAS the system generates a lower load on the downlink channel, reaching around one third of the channel occupation compared to SE-CAS. This is due to how the system manages the DENMs dissemination. As explained in Section IV, in this case the DENMs are used to stop and start the vehicles, yielding a more efficient usage of the downlink channel. By looking at the values for increasing vehicle spatial density, it is possible to see a linear increase in the channel load. Due to the different way in which the vehicle mobility is managed by the system, and due to the way in which the system generates DENMs, the increase in the vehicle maximum speed is always associated to an increase of the downlink channel load.

Latency. In our framework, every time the client eCA application sends a CAM, it includes a timestamp in the message. The timestamp is used at the server side to derive the client-to-server latency, namely, the uplink latency shown in Figure 9. The same is done when the server generates a DENM, allowing the client to calculate the downlink latency.

In this case, the results are averaged over a single simulation campaign, 50 simulations, 300 seconds of length, with a density of 4 vehicles per kilometer and speed fixed to 27.78 m/s.

Figure 9 shows how the uplink channel contributes with 12 ms of delay, while in downlink the value is around 4.5 ms. These values generate a final RTT latency of 16.4 ms. The results are quite stable, with a standard deviation over the whole data set of 8.5×10^{-8} in uplink and of 8.5×10^{-4} in downlink. The delay is due to how the MAC-to-channel delay is modeled in the simulator (MAC-to-channel delay is configured to 4 ms in uplink and 2 ms in downlink) and when the scheduling decision is made by the eNB scheduler (that can add around 1 or 2 extra milliseconds). In addition to this, in uplink, the UE notifies the eNB that it has data to send. Then, the eNB assigns some resources to the UE and, finally, the UE sends the data through the data channel. Therefore, there are three transmissions at the physical layer: two in the control channel and one in the data channel. Those values are in line with the latency requirements for V2X applications, which normally range between 10 and 100 ms [36], [37].

PDR. The PDR was studied by increasing the maximum speed, from 8.33 m/s to 36.11 m/s, and can be seen in Table 3. The results show a PDR always higher than 0.99, with very small variation around the average value. The results seem to be unaffected by the presence of the fading model, indeed we do not observe any correlation between the PDR and the increase of the vehicle maximum speed.

Collisions Avoided. The results for this metric of interest have been collected by varying speed, spatial density, CAS deployed, and reaction time.

In Figure 10, it is possible to see the percentage of collisions avoided when the SE-CAS mechanism is deployed, and the vehicle drivers wait for 1 second before starting the braking phase after the DENM is received at application layer. These results have been collected by comparing the case in which the eCA service is active, to the case in which it is not active, by keeping the same mobility traces for the two sets of simulations. For very low speeds, we achieve an ideal behavior (i.e., 100% of collisions avoided); however, for higher speeds, the results show a clear correlation between the vehicle maximum speed, and the percentage of collisions avoided: by increasing the speed, it will be harder for vehicles to stop, and the algorithm tends to warn the vehicles too late to avoid the collision. Another parameter that has a direct impact on the final results, is the vehicle spatial density: the

TABLE 3: Packet Delivery Ratio (PDR) in uplink and downlink.

Speed [m/s]	PDR - Average uplink /downlink	PDR - Std Dev uplink/downlink
8.33	0.99996 / 0.99917	$3.28 \times 10^{-5} / 5.08 \times 10^{-3}$
13.89	0.99997 / 0.99976	$2.24 \times 10^{-5} / 1.60 \times 10^{-3}$
20.83	0.99996 / 0.99627	$2.67 \times 10^{-5} / 1.29 \times 10^{-2}$
27.78	0.99997 / 0.99628	$2.73 \times 10^{-5} / 7.3 \times 10^{-3}$
36.11	0.99997 / 0.99641	$4.51 \times 10^{-5} / 1.05 \times 10^{-2}$

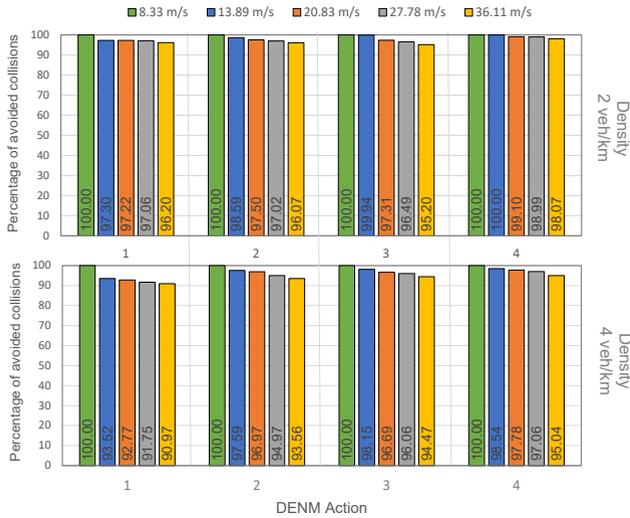


FIGURE 10: Percentage of collisions avoided with SE-CAS, reaction time set to 1 s. In the x axis the different DENM actions are plotted, with each bar corresponding to a different speed. The two plots correspond to different vehicle spatial densities.

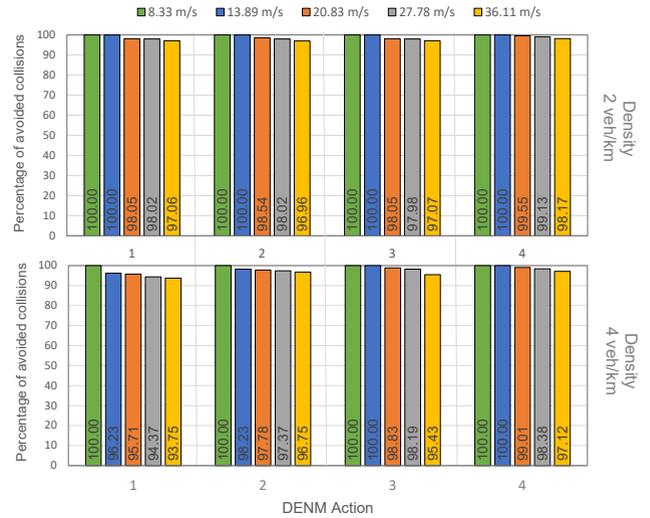


FIGURE 12: Percentage of collisions avoided with SE-CAS, reaction time set to 0.05 s. In the x axis the different DENM actions are plotted, with each bar corresponding to a different speed. The two plots correspond to different vehicle spatial densities.

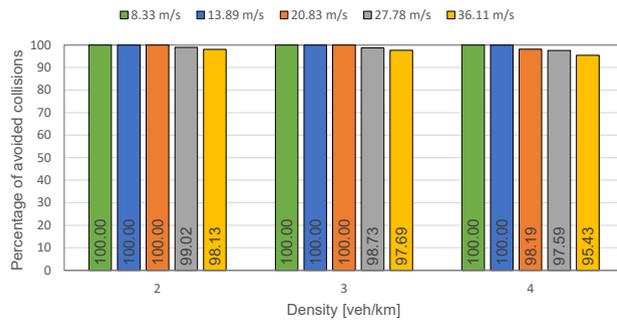


FIGURE 11: Percentage of collisions avoided with ME-CAS, reaction time set to 1 s. In the x axis, the vehicle spatial density is plotted; each bar correspond to a different speed.

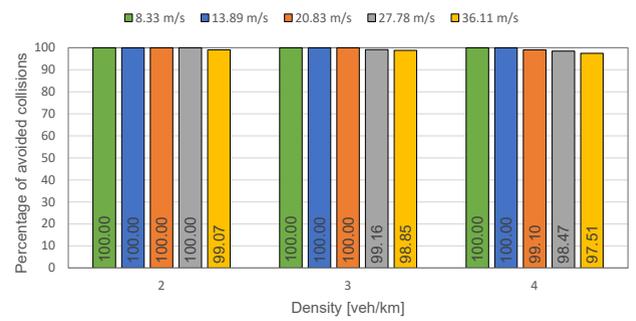


FIGURE 13: Percentage of collisions avoided with ME-CAS, reaction time set to 0.05 s. In the x axis, the vehicle spatial density is plotted; each bar correspond to a different speed.

higher it is, the lower the final number of collision avoided, due to the more challenging, complicated configurations that may rise in a crowded intersection.

The results when deploying the ME-CAS mechanism are shown in Figure 10, still in the case of a human driver with 1 s-reaction time. Here, the x axis represents the vehicle spatial density, and we can see again a visible direct relationship between the number of vehicles present in the scenario and the final number of collision avoided. Moreover, the correlation, between the vehicle maximum speed and the application capability to avoid collisions, still holds.

When adopting ME-CAS we can see a general improvement of the results, in terms of collisions avoided. This happens because the algorithm managing the vehicle mobility, in this case, has a wider perception of the intersections status, by considering multiple collision events at the same time.

The results when setting the reaction time to 0.05 s are shown in Figure 12 and Figure 13. In this case, the vehicles

are assumed to have an automatic braking system, which starts the vehicle deceleration after 0.05 s since the DENM₀ was received. Looking at the SE-CAS behavior, one can observe that action 4 outperforms 3 and 2, which in turn provide slightly better results than action 1, especially under higher vehicle densities. This is due to the fact that action 4 allows for more time to vehicles to stop, hence reducing the number of new, potential collision situations. Therefore, for the SE-CAS and the scenario under evaluation, action 4 is the best choice.

In general, by increasing the reactivity of the system, we leave more braking room to vehicles and avoid collisions. In our scenario this translates into a higher number of collisions avoided, using both SE-CAS and ME-CAS.

The increase of the vehicle spatial density and maximum speed are still associated with an increased number of collisions not avoided, albeit not so high as in the case of human drivers. Again, deploying ME-CAS is beneficial in terms of

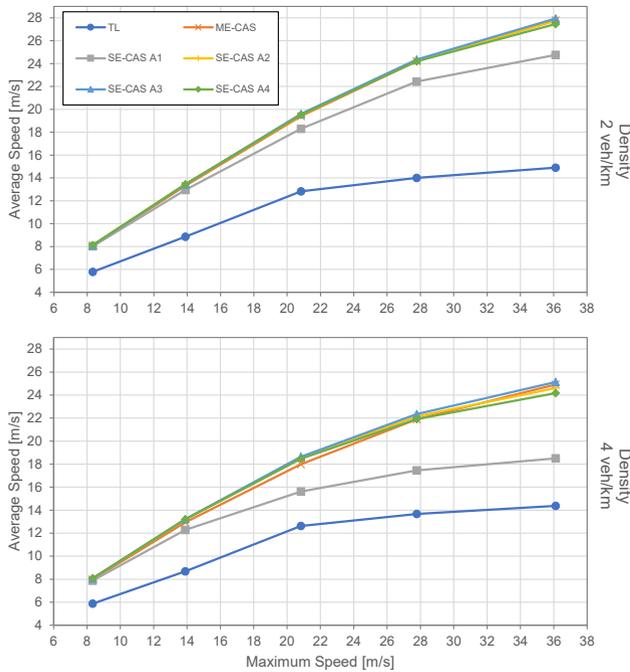


FIGURE 14: Average speed kept by vehicle, as a function of their theoretical maximum speed. Each line corresponds to a different scenario (with or without eCA), while the two plots correspond to different vehicle spatial densities.

collision avoided, reaching 100% of collisions avoided for all densities and speed up to 13.89 m/s.

At last, we highlight that, to our knowledge, the solution we propose is the first one entirely based on CAM and DENM dissemination. The solutions that car makers already deploy in their vehicles are based on algorithms involving cameras or lidars and, most importantly, are proprietary, hence not publicly available¹.

Speed. To compare the average speed results obtained by deploying the eCA service with a classic scenario, an additional set of simulations was performed, to study the average speed when regular traffic lights are deployed at the two intersections. Note that we only report here a comparison in terms of traffic flow speed, since statistics are already available from real-world studies. For instance, NHTSA reports that more than 50% of all the intersection-related crashes happen in crossings regulated by traffic lights [23].

In the scenario we consider, the two traffic lights follow a deterministic scheme, with a cycle time of 90 s. Since the vehicular traffic is balanced on all arms of the intersection, a smarter traffic light algorithm would not yield significant benefits in terms of traffic flow efficiency. The results are shown in Figure 14, with the two plots corresponding to different spatial densities and each line corresponding to a different scheme. The results suggest that the spatial density has an impact on the final average speed: the higher the density, the lower the measured average speed. Indeed, in the

¹Ours can be seen as a worst-case analysis with respect to the case where proprietary solutions are adopted.

case in which the eCA service is deployed, the increasing density leads to a higher number of generated DENMs, which corresponds to a higher number of vehicles stopped. In the scenario with the traffic light, increasing the spatial density corresponds to the lengthening of the queues created at intersections. Consequently, the vehicles will take longer to restart, resulting in a decrease of the overall average speed. Note that, however, in the traffic light scenario such a decrease is marginal when compared to the speed reduction observed under eCA as the vehicle density doubles.

A look at the evolution of the plot under different scenarios (SE-CAS, ME-CAS, and traffic light), shows how a service like eCA can drastically improve the mobility efficiency. However, in ME-CAS this is achieved with around one third of the downlink traffic of SE-CAS, which reflects the better global knowledge acquired by ME-CAS. This result makes ME-CAS a promising solution to handle much more complex scenarios. Finally, we observe that, SE-CAS and action 1 (i.e., the version that stops both vehicles on collision course), the overall average speed is lower, but still higher with respect to the scenario managed by the traffic lights.

IX. CONCLUSIONS

A system that avoids vehicle collisions in regulated and unregulated intersections can potentially save millions of lives. In this paper, we have proposed an enhanced Collision Avoidance (eCA) service deployable in the cellular edge that collects vehicle positional beaconing and detects impending collisions, informing the involved vehicles and providing them with hints on evasive actions. In case multiple vehicles are involved in the same potential collision event, a contention procedure collectively manages the right-of-way of those vehicles, reducing the resulting downlink data traffic needed to alert them. Using a simulation framework that combines SUMO and ns-3, we have designed and compared solutions with different levels of complexity. We found that our schemes perform very well: most potential collisions are avoided (their likelihood being inversely proportional to the vehicle speed and spatial density, as expected), while the vehicle traffic retains a better fluidity than in a case where the intersection is regulated by traffic lights. At last, we remark that, in order to be effective, our solution requires a fairly high technology penetration rate, since a collision can be avoided only when both involved vehicles are connected cars.

Future work will address two main aspects: (i) the assessment of the number of false positives in the eCA system, and (ii) the study of larger-scale scenarios with higher vehicles densities and multiple eCA servers.

...

REFERENCES

- [1] World Health Organization, "Global status report on road safety 2018," 2018, <https://apps.who.int/iris/bitstream/handle/10665/276462/9789241565684-eng.pdf>, Last accessed on 11-11-2019.
- [2] "EU H2020 5G-PPP Project 5G-TRANSFORMER," <http://5g-transformer.eu>, Last accessed on 27-11-2019.

- [3] J. Baranda, G. Avino, J. Mangués, L. Vettori, R. Martínez, C. F. Chiasserini, C. Casetti, P. Bande, M. Giordanino, and M. Zanzola, "Automated deployment and scaling of automotive safety services in 5G-Transformer," in 5th IEEE Conference on Network Functions Virtualization and Software Defined Networking (IEEE NFV-SDN 2019), Dallas, Texas (USA), 2019.
- [4] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward Efficient Content Delivery for Automated Driving Services: An Edge Computing Solution," *IEEE Network*, vol. 32, no. 1, pp. 80–86, Jan 2018.
- [5] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling Collaborative Edge Computing for Software Defined Vehicular Networks," *IEEE Network*, vol. 32, no. 5, pp. 112–117, Sep. 2018.
- [6] G. Avino, P. Bande, P. A. Frangoudis, C. Vitale, C. Casetti, C. F. Chiasserini, K. Gebru, A. Ksentini, and G. Zennaro, "A MEC-based Extended Virtual Sensing for Automotive Services," *IEEE Transactions on Network and Service Management*, 2019.
- [7] L. Li, Y. Li, and R. Hou, "A Novel Mobile Edge Computing-Based Architecture for Future Cellular Vehicular Networks," in 2017 IEEE Wireless Communications and Networking Conference (WCNC), March 2017, pp. 1–6.
- [8] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 94–100, July 2017.
- [9] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in 2017 IEEE International Conference on Communications (ICC), May 2017, pp. 1–6.
- [10] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: a survey," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 148–157, May 2013.
- [11] S. Kato, M. Hiltunen, K. Joshi, and R. Schlichting, "Enabling Vehicular Safety Applications over LTE Networks," in 2013 International Conference on Connected Vehicles and Expo (ICCVE), Dec 2013, pp. 747–752.
- [12] "EU H2020 5G-PPP Project 5GCAR, 5G Communication Automotive Research and innovation," <http://5gcar.eu>, Last accessed on 27-11-2019.
- [13] A. Mukhtar, L. Xia, and T. B. Tang, "Vehicle Detection Techniques for Collision Avoidance Systems: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2318–2338, Oct 2015.
- [14] Q.-H. Nguyen, M. Morold, K. David, and F. Dressler, "Car-to-Pedestrian communication with MEC-support for adaptive safety of Vulnerable Road Users," *Computer Communications*, 11 2019.
- [15] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1204–1216, July 2017.
- [16] M. Yousef, A. Hosny, W. Gamil, M. Adel, H. M. Fahmy, M. S. Darweesh, and H. Mostafa, "Dual-Mode Forward Collision Avoidance Algorithm Based on Vehicle-to-Vehicle (V2V) Communication," in 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), Aug 2018, pp. 739–742.
- [17] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino, and S. Scarpina, "Performance Analysis of C-V2I-Based Automotive Collision Avoidance," in 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), June 2018, pp. 1–9.
- [18] G. Avino, M. Malinverno, C. Casetti, C. F. Chiasserini, F. Malandrino, M. Rapelli, and G. Zennaro, "Support of Safety Services through Vehicular Communications: The Intersection Collision Avoidance Use Case," in 2018 International Conference of Electrical and Electronic Technologies for Automotive, July 2018, pp. 1–6.
- [19] M. Malinverno, G. Avino, C. Casetti, C. F. Chiasserini, F. Malandrino, and S. Scarpina, "MEC-based Collision Avoidance for Vehicles and Vulnerable Users," *IEEE Vehicular Technology Magazine*, 2019.
- [20] Y. Zhang, X. Yan, X. Li, J. Wu, and V. V. Dixit, "Red-Light-Running Crashes Classification, Comparison, and Risk Analysis Based on General Estimates System (GES) Crash Database," in *International journal of environmental research and public health*, 2018.
- [21] A. Jahangiri, H. Rakha, and T. A. Dingus, "Red-light running violation prediction using observational and simulator data," *Accident Analysis and Prevention*, vol. 96, pp. 316 – 328, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0001457516302056>
- [22] N. Kim, J. Kim, H. Kim, K. Lim, Y. Ko, N. Jeong, A. H. Smith, and H. A. McNally, "Red Light Running Prediction System using LIDAR," in 2019 IEEE Sensors Applications Symposium (SAS), March 2019, pp. 1–5.
- [23] Dot Hs, "Crash Factors in Intersection-Related Crashes: An On-Scene Perspective," 2010.
- [24] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34. [Online]. Available: https://doi.org/10.1007/978-3-642-12331-3_2
- [25] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, "An Open Source Product-oriented LTE Network Simulator Based on Ns-3," in *Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. New York, NY, USA: ACM, 2011, pp. 293–298. [Online]. Available: <http://doi.acm.org/10.1145/2068897.2068948>
- [26] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic Traffic Simulation using SUMO," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Nov 2018, pp. 2575–2582.
- [27] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: An Interface for Coupling Road Traffic and Network Simulators," in *Proceedings of the 11th Communications and Networking Simulation Symposium*, ser. CNS '08. New York, NY, USA: ACM, 2008, pp. 155–163. [Online]. Available: <http://doi.acm.org/10.1145/1400713.1400740>
- [28] ETSI, "ETSI 302 637-2 - v1.3.1," *European Telecommunications Standards Institute, Tech. Rep.*, 2014.
- [29] —, "ETSI 302 637-3 - v1.3.1," *European Telecommunications Standards Institute, Tech. Rep.*, 2019.
- [30] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: a survey," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 148–157, May 2013.
- [31] H. Summala, "Brake Reaction Times and Driver Behavior Analysis," *Transportation Human Factors*, vol. 2, pp. 217–226, 09 2000.
- [32] M. Staubach, A. Kassner, N. Fricke, C. Schießl, M. Brockman, and D. Kuck, "Driver reactions on ecological driver feedback via different HMI modalities," in *Vorgestellt in: Proceedings of the 19th World Congress on ITS*, Wien Österreich, 2012.
- [33] M. Green, "How long does it take to stop? Methodological analysis of driver perception-brake times," *Transportation Human Factors*, vol. 2, no. 3, pp. 195–216, 2000.
- [34] S. Lin, Y. Zhang, C. Hsu, M. Skach, M. Haque, L. Tang, and J. Mars, "The Architectural Implications of Autonomous Driving: Constraints and Acceleration," 03 2018, pp. 751–766.
- [35] R. Prohaska and P. Devlin, "Combined Brake and Steering Actuator for Automatic Vehicle Control," no. qt9366m38b, Aug. [Online]. Available: <https://ideas.repec.org/p/cdl/itsrrp/qt9366m38b.html>
- [36] Z. Amjad, A. Sikora, B. Hilt, and J. Lauffenburger, "Low Latency V2X Applications and Network Requirements: Performance Evaluation," in 2018 IEEE Intelligent Vehicles Symposium, June 2018, pp. 220–225.
- [37] ETSI, "ETSI TS 122 185 - V14.3.0," *European Telecommunications Standards Institute, Tech. Rep.*, 2017.



MARCO MALINVERNO (M'19) received the B.S. degree in telecommunication engineering from Università degli Studi di Siena, Italy, in 2014 and the M.S. degree in computer and communication networks engineering from Politecnico di Torino, Italy, in 2016. He is currently pursuing the Ph.D. degree in connected vehicles for autonomous driving at Politecnico di Torino, Italy.

From 2016 to 2017, he was a researcher at the Department of Electronics and Telecommunications, Politecnico di Torino, Italy, where he worked on containerized software solutions for vehicular networks. His research interests include the development of safety applications for connected cars, the implementation of vehicular communications using off-the-shelf hardware, and the performance analysis of vehicular networks.



MANUEL REQUENA-ESTESO is M.Sc. in computer science from the Technical University of Valencia (UPV, 1998). He realized his master thesis in the Ecole Nationale Supérieure de l'Electronique et des Applications (ENSEA) in France. He worked as Software Engineer developing Telecommunication Software Solutions (Atos Origin Integration, France, 1998 - 2003). Currently, he is a Senior Research Engineer and the coordinator of the EXTREME Testbed at Centre

Tecnològic Telecomunicacions Catalunya (CTTC), Barcelona and the responsible of the laboratory in the Mobile Networks Department of the CTTC. Moreover, he is one of the designers, developers and maintainers of the LTE module of the ns-3 simulator.

His current research interests include design and implementation of networking protocols for IP networks, self-organized wireless mesh networks, cellular networks (LTE, HSPA, UMTS), all-IP heterogeneous environments, Network Function Virtualization (NFV), Software-Defined Networking (SDN) and networks simulation.



JOSEP MANGUES-BAFALLUY is Senior Researcher and Head of the Communication Networks Division of the Centre Tecnològic Telecomunicacions Catalunya (CTTC), Barcelona. He has participated in various roles (incl. leadership) in several public funded and industrial research projects (e.g., 5GPPP 5Growth, 5G-Transformer, or Spanish 5G-REFINE). He was vice-chair of IEEE WCNC 2018 (Barcelona). Previously, he was also researcher and assistant professor at

UPC, from which he received the degree and PhD in Telecommunications Engineering in 1996 and 2003, respectively.

Research interests: NFV applied to mobile networks and autonomous network management.



CLAUDIO ETTORE CASETTI is an Associate Professor at the Department of Control and Computer Engineering, Politecnico di Torino, Italy. He has published more than 200 papers in peer-reviewed international journals and conferences on the following topics: vehicular networks, 5G networks, transport and network protocols in wired networks, IEEE 802.11 WLAN. According to Google Scholar, his H-index is 38. He has been in the organization and technical committees of

the major IEEE/ACM conferences (IEEE INFOCOM, IEEE CCNC, IEEE Globecom, ACM Sigmetrics...). He has given Tutorials on vehicular networks at major IEEE Conferences, including IEEE ICC, IEEE Globecom, IEEE CCNC and IEEE VTC. He is a Senior Member of IEEE.



JORGE BARANDA is currently a Senior Researcher in the Mobile Networks department at Centre Tecnològic Telecomunicacions Catalunya (CTTC), Barcelona, and Senior Member of IEEE. He received an M.S. degree in Electrical Engineering from the Technical University of Catalonia in 2008. At CTTC, he has participated in several European, national and industrial projects related with management and orchestration of SDN/NFV mobile networks, efficient routing strategies for

mobile network backhauling and novel wireless communication systems. He has co-authored over 30 different peer-reviewed journal and conference papers.

His current research interests include management and orchestration of SDN/NFV mobile networks, wireless communications, wireless backhaul, network routing protocols, and network optimization.



CARLA FABIANA CHIASSERINI (M'98, SM'09, F'18) received her Ph.D. from Politecnico di Torino, Italy, in 2000. She worked as a visiting researcher at UCSD and as a Visiting Professor at Monash University. She is currently a Professor with the Department of Electronic Engineering and Telecommunications at Politecnico di Torino.

Her research interests include architectures, protocols, and performance analysis of wireless networks.