Computer Networks 172 (2020) 107150

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Collision avoidance in 5G using MEC and NFV: The vulnerable road user safety use case

Sokratis Barmpounakis*, George Tsiatsios, Michael Papadakis, Evangelos Mitsianis, Nikolaos Koursioumpas, Nancy Alonistioti

Department of Informatics and Telecommunications, National and Kapodistrian, University of Athens, Athens, Greece

ARTICLE INFO

Keywords:
5G
Collision avoidance
Multiple access edge computing
Mobile edge computing
Network function virtualization
Trajectory prediction

ABSTRACT

Automotive is considered one of the driving use cases for the 5th Generation (5G) systems, which currently formulates numerous scenarios and Key Performance Indicators (KPIs), via advanced Vehicle-to-everything (V2X) services and applications. Minimum end-to-end delay, as well as advanced contextual awareness requirements, pose novel architectural and functional challenges. This paper exploits two key enablers, namely Multiple Access/Mobile Edge Computing (MEC) and Network Function Virtualization (NFV), and acts in a two-fold manner: Firstly, it proposes a hybrid architecture for 5G systems, which exploits the afore-mentioned technologies, and performs computing resources' selection among MEC and/or centralized, cloud-based resources (as VNFs), towards efficient service orchestration. The second contribution of this paper is a novel V2X service and algorithm, namely VRU-safe, that operates on top of the proposed architecture. VRU-Safe is an efficient, lightweight, low time complexity scheme, capable of identifying and predicting potential imminent road hazards between moving vehicles and Vulnerable Road Users (VRUs). The performance and viability of the proposed solutions are evaluated in a real-world 5G testbed in Europe.

1. Introduction

Fifth Generation (5G) networks promise highly flexible and programmable end-to-end communication aiming to further enhance several network performance aspects. Increased performance in terms of throughput, latency and reliability is expected in order to meet challenging requirements from diverse services and user demands [1]. Among 5G use cases, the automotive vertical domain establishes an undoubted key driver for 5G systems. Connected and fully automated vehicles will ultimately lead to safer transportation via ultra-low delay network performance with the support of the roadside infrastructure, aiming at radically reducing fatalities on the road, while at the same time achieving lower environmental impact [2].

Very recently, the 5G Infrastructure Public Private Partnership (5G-PPP) released an initial strategic agenda for Connected and Automated Mobility (CAM) in Europe [3]. In this agenda, key elements are highlighted in order to progress and stimulate investments into a pan-European network of 5G corridors, as a first strategic step towards large scale deployment and other high value services related to connected vehicles. Vehicle-to-everything (V2X) communication enablers have been introduced by 3GPP in Long Term Evolution (LTE) since Release 14 [4], via the Cellular V2X concept (i.e., C-V2X), - as well as its successor set of features, i.e., eV2X -, in several 5G use cases [5,6], as well as by the European Telecommunications Standards Institute (ETSI) [7]. 3GPP aims to further enhance several aspects of V2X communications in Release 16 [8]. In particular, collision avoidance, VRU safety and hazardous situation detection is identified as one of the key use cases according to 5G Automotive Association (5GAA) [9]. Additionally, the requirements for different V2X use cases has been identified in [10], in relation to the end-to-end latency, reliability, data rate per vehicle, communication range, etc. In the same work, the specific requirements for the VRU, Vehicle-to-Pedestrian use case (V2P) have been identified as well, highlighting a maximum end-to-end latency of 100 ms, reliability higher than 95%, and data rate ranging between 5 and 10 kb/s. Also, in [11], the authors provide a comprehensive analysis on the V2X requirements according to ETSI and 3GPP, and focus on a thorough discussion on the 5G requirements, particularly for the Radio Access.

Towards this direction, 5G networks call for the support of multiple network slices on a common and programmable infrastructure, i.e. multiple logical networks with different configurations, optimized for specific traffic service types. Recent research efforts target to the design of 5G network slice(s) customized for vehicle-to-everything services, which involve vehicles exchanging data with each other, with

* Corresponding author.
E-mail addresses: sokbar@di.uoa.gr (S. Barmpounakis), georget@di.uoa.gr (G. Tsiatsios), mpapadakis@di.uoa.gr (M. Papadakis), evmitsia@di.uoa.gr (E. Mitsianis), nkoursioubas@di.uoa.gr (N. Koursioumpas), nancy@di.uoa.gr (N. Alonistioti).

https://doi.org/10.1016/j.comnet.2020.107150
Received 23 November 2019; Received in revised form 22 January 2020; Accepted 10 February 2020
Available online 2 March 2020

the infrastructure and any communicating entity for improved transport fluidity, safety, and comfort on the road [12]. The potential of the network slicing concept for V2X has been also initially investigated by 3GPP [13] and also unveiled in [14,15].

Although automotive-driven services involve different traffic types, each with diverse requirements, -primarily in terms of end-to-end latency, reliability and bandwidth-, these requirements could be clustered in two major kinds of applications: (a) large scale applications, that involve non delay-sensitive communications between road users and application servers located remotely, such as fleet management or traffic monitoring; (b) time critical applications, that deal with short-lived information (hence time-critical), such as vehicles' dynamic parameters and sensor data, relevant in proximity of the area where this data been generated. To this end, its treatment in proximity to the users is also required. Anticipated Collision Avoidance is a typical use case requiring this kind of time-critical information processing.

Virtualized cloud resources offer robust performance, via powerful processing and computing capabilities. On the other hand, MEC capabilities are introduced towards the enablement of placing storage and computation resources at the network edge, in the proximity of the radio access network. By processing data locally and accelerating data streams through various techniques, MEC reduces both the end-to-end latency, as well as the traffic overhead towards the core network. A crucial trade-off, thus, results between the two architectural choices, with MEC-based processing, on the one hand, minimizing end-to-end communication delay; on the other hand, minimizing the processing/computing delays via robust and powerful virtualized cloud resources, having however to face higher transmission end-to-end delays between the RAN and the cloud.

The remainder of this article is organized as follows. Section II presents the state of the art. In Section III, we present in detail the proposed architecture and the VRU-Safe service. Section IV provides the performance evaluation-related details, the experimental set-up and discusses the obtained results. The article concludes in the final section, with a summary recapping the main findings and discussing the next steps of this work.

2. Related work

Recently, MEC-based solutions empowered the automotive sector facilitating the direct exchange of vehicle related information between (mobile) nodes via the underlying communication network, highlighting novel challenging use cases and applications [16–19]. Resource-heavy applications will leverage from edge computing characteristics, by minimizing computational cost, latency and supporting data processing strategies [20]. Besides the end-to-end delay reduction, MEC-based architectures assist towards achieving higher scalability [21–24]. The authors in [25] survey existing concepts integrating MEC functionalities to the mobile networks and discuss current advancement in standardization aspects of MEC. The survey focuses on user-oriented use case in the MEC, i.e., computation offloading and divides related research to three key areas: i) decision on computation offloading, ii) allocation of computing resource within the MEC, and iii) mobility management.

In [26], an architecture for collision avoidance systems is proposed. The collision detection takes place via Collision Detectors, i.e., computing entities, such as physical servers, virtual machines or containers, that run a collision-detection algorithm and may be deployed in different parts of the network, such as MEC servers, or close to the network core. The authors present in detail the collision detection scheme and prove its validity and performance it via an urban area-based simulation environment.

In [27], the authors propose a VRU warning system, which aims to alert road users (e.g. pedestrians, cyclists, vehicles) about the presence of nearby moving users, in case of hazardous situations. The authors present a simple architecture that consists of a CAM client and a CAM server, responsible for transmitting the respective notifications the users. The proposed prediction approach is a simple, threshold-based evaluation of the distance between the VRU and the vehicle, while the mechanism is evaluated by a minimal experiment with one scenario that compares Wi-Fi, MEC-enabled LTE and traditional, core cloud-based LTE communication.

A 5G/LTE-based protection system for VRUs is described in [28], which introduces a "context-filter" capable of identifying VRUs in potentially dangerous situations based on several types of contextual information (VRU position, movement direction, accelerations). Particularly, the authors propose the exploitation of smart phone sensor data in order to augment their model. The evaluation of the propose scheme generate interesting trends in relation to different scenarios, such as different crossing pedestrian curb heights.

The improvement implied by MEC for Vehicle-to-Vulnerable Road Users communications was evaluated in [29] via an extensive simulation-based performance comparison between the conventional and the MEC-assisted network architecture, showcasing considerable latency reduction results in the MEC approach, comparing with centralized cloud-based systems.

Exploiting high-performance computing and storage facilities, as well as a diversity of in-vehicle deployed sensors, vehicles, MEC-based solutions will be able to process massive amounts of information-rich data towards valuable knowledge extraction. In several works, offloading vehicle-oriented big data to the MEC server for preprocessing is proposed in the context of the Internet of Vehicles (IoV), towards several services, such as pricing, entertainment and multimedia applications, and vehicle sensor data analytics towards the users [30,31]. V2X sector can largely benefit from Machine Learning (ML) methods, such as deep learning and deep reinforcement learning [32,33] and create more intelligent V2X networks. ML techniques can leverage from the data generation, thereby making predictions for effective autonomous driving strategies, suitable for real-time embedded systems [34] targeting efficient, comfortable, and safer driving. Diverse simulations [35], real-world experiments [36] or combination of both approaches [37] open new horizons towards road safety.

Nevertheless, MEC capabilities - if not combined with virtualization enablers - will be limited, due to the lack of flexibility of the current network, which is based on specific-purpose hardware. In order to overcome those barriers, SDN and NFV technologies are going to support such solutions, by providing high flexibility, dynamic NFV placement and service orchestration, higher reliability, as well as enhanced, responsive resource allocation approaches [38–40]. The afore-presented solutions mainly take advantage of the low latency, elasticity, scalability of the MEC to address specific functional requirements of the automotive sector while overcoming typical cloud issues. To this end, flexible and agile edge communications open new opportunities for future applications to support Road Safety.

Besides the innovation introduced in architectural or virtualization aspects, other works focus on protocol enhancements. In [41], the authors propose a method for enhancing cooperative driving via full-duplex radios, in IEEE 802.11 radio access technology. More specifically, an enhanced carrier-sense multiple access with collision avoidance (CSMA/CA) scheme is described, which improves the CAM timeliness and enables full-duplex devices to listen to the channel while transmitting. Via an extensive evaluation, the authors prove the validity and viability of the proposed solution for different V2X scenarios.

In this work, we propose a novel architecture that performs context-based resource selection between MEC- and cloud-based computing resources; on top we propose and evaluate an efficient, low complexity collision avoidance service for VRUs. A key novelty is that the proposed solution – contrary to the existing literature - does not attempt to choose between a cloud-based or a MEC-based solution as superior, as this is directly linked and dependent to the specific scenario, as well as the particular infrastructure specifications and resource availability. To this end, the proposed architecture demonstrates the methodology and algorithms required in order to exploit both architectural choices, in an efficient and intelligent manner, applicable to diverse set-ups. To our

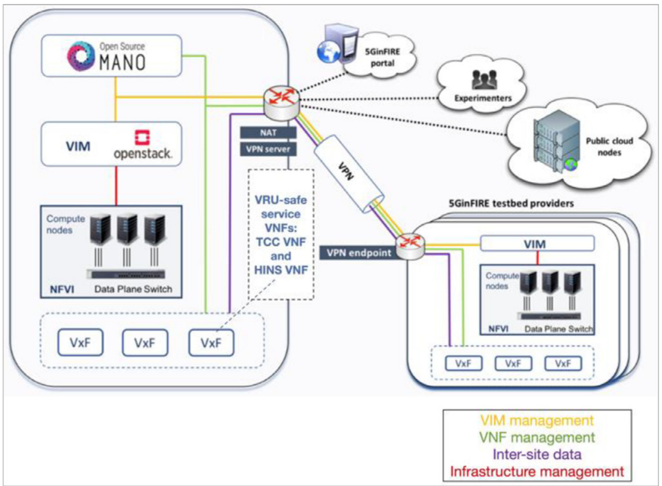


Fig. 1. The VRU-safe 5GinFIRE MANO-based architecture.

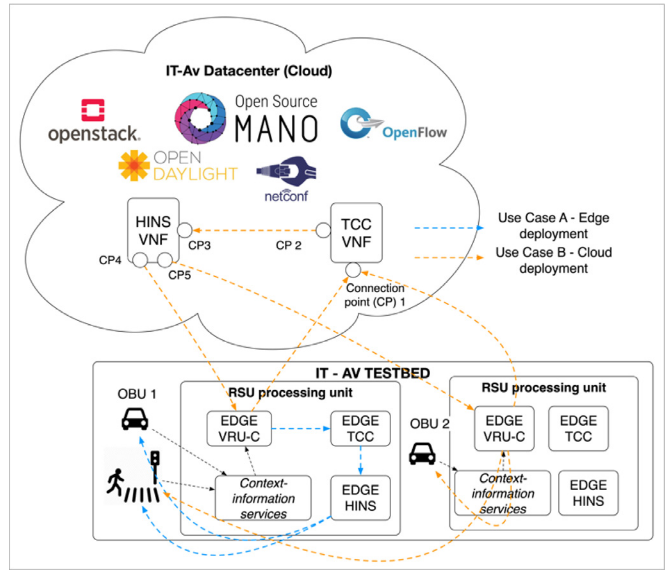


Fig. 2. VRU-Safe service architecture and overview.

knowledge, there is no previous work in the domain of V2X services and applications for 5G architectures, which performs such a dynamic, context-based, hybrid MEC/Cloud resource selection approach. Additionally, the proposed collision avoidance algorithm is a novel, lightweight and low time complexity solution, capable of performing efficient trajectory prediction and collision avoidance, towards adequately addressing the critical 5G requirements, in terms of end-to-end delay, high reliability and scalability. Last but not least, no previous work has been evaluated to such an extent, in a real network set-up, exploiting well-established virtualization and service orchestration frameworks, such as the ETSI Open Source MANO, and with the involvement of real, human-operated vehicles, via a diverse and realistic set of scenarios.

3. The proposed architecture and collision avoidance service

The proposed service, namely VRU-Safe, aims at timely predicting and avoiding anticipated collision events between VRUs and vehicles in their vicinity; at the same time, it builds on top of a novel architecture towards 5G, capable of dynamically steering the computing requests towards either MEC or centralized cloud computing resources (operating as VNFs), towards efficient V2X service operation and delay minimization for critical scenarios.

The end-to-end design, deployment and orchestration of the described network service took place in the context of a mature 5G testbed federation, and specifically based on the 5GinFIRE ETSI MANO platform (Fig. 1), for the cloud-based operation.

The primary role of the 5GinFIRE ETSI MANO framework is to orchestrate the VRU-safe services and VNFs towards the end-to-end VRU-safe service operation chain. Each VNF is described by a virtual function descriptor (VNFD). Both of the VNFDs must be encapsulated in a Network Service Descriptor (NSD), which is a service, describing the components of the aforementioned network service. The NSD also exposes a set of points in order to enable the connectivity between the relevant VNFs, Virtual Link elements or the external network. During the actual experimentation, a mirror platform from the 5GINFIRE MANO was used, which replicates the functional features of the main platform. Additionally, towards the enablement of remote orchestration, management and execution capabilities, Virtual Private Networking (VPN) features are also provided by the 5GinFIRE architecture.

The main components of the framework are a Trajectory Computing Component (*TCC*), a Hazard Identification and Notification Service (*HINS*) and the VRU-Safe Controller (*VRU-C*), which is the primary control module of the service; VRU-C evaluates the MEC/Cloud computing resources selection, i.e., whether the service needs to run locally at the MEC servers, or be forwarded to the cloud for VNF-based processing. The VRU-safe MANO platform deployment comprises 2 VNFs, i.e., the TCC VNF and the HINS VNF. The rest of the components are operating locally, on the MEC side.

3.1. THE VRU-C controller module

VRU-C module provides the core logic of the service, which analyzes the context of the network environment, i.e. a) the location and mobility characteristics of the involved UEs/entities (OBUs, VRUs, etc.), b) their associated radio access points of attachment and c) the computing resources' availability (in terms of utilization load), in order to decide which TCC and HINS processes will be triggered: a) the MEC computing mode of the system, i.e., the TCC and HINS components, which are running locally, on the road infrastructure as plain software deployed in the RSUs, or b) the virtualized network services running on the cloud-based, Network Function Virtualization Infrastructure (NFVI), in the form of Virtual Network Functions (VNFs).

The high-level architecture is illustrated in Fig. 2:

The main algorithmic steps of VRU-C are provided in Algorithm 1 pseudocode.

The main rationale behind this approach is directly linked to the main objective of the VRU-safe architecture, which is the timely prediction of critical situations; whenever possible (i.e. MEC resources are sufficient and context/mobility information of the involved OBUs/VRUs is available), the proposed system performs the required processing and notification triggering as close as possible –both in geographical terms, as well as in terms of network hops- to the infrastructure, i.e., the network edge, avoiding redundant signaling overhead, between the RSU and the Cloud. When both the involved OBU and VRU are connected to the same RSU (e.g. via 802.11p), the required context information for the TCC and HINS processing is directly available in a common MEC location; as a result, as long as the available computing resources of the specific RSU are sufficient, the information is directly processed (via the MEC-based TCC and HINS modules) and potential alerts are transmitted back to the devices (OBU, VRU) via the southbound interface.

Although similar architectures have been proposed for 5G networks and V2X applications introducing MEC solutions, the novelty of the particular architecture lies (a) on the control module, namely the VRU-C, that is applied at the MEC side and performs resource selection among MEC and cloud computing resources and (b) on the low time complexity approach that is described in the following subsections for the collision

Algorithm 1 VRU_C algorithmic steps in pseudocode.

Variables $i, j, k \in N$;
Initialize lists for the $RSU - VRU$ *ans* $RSU - OBU$ *association*;
$RSU_{k_{VRU}}[] \leftarrow 0$, $RSU_{k_{OBU}}[] \leftarrow 0$;
  Initialize pointers for K RSUs $k \leftarrow 0$;
  while $k < K$ *do*
  Initialize pointers for M VRUs, N OBUs: $i \leftarrow 0, j \leftarrow 0$;
  while $i < M$ do
  MEC VRU_C: *process context information from* VRU_i;
    Store association information to $RSU_{k_{VRU}}[]$;
    $i++$;
  end
  while $j < N$ do
  MEC VRU_C: *process context information from* VRU_i;
    Store association information to $RSU_{k_{OBU}}[]$;
    $j++$;
  end
end
$k \leftarrow 0$;
for each $k < K$ *do*
  if $RSU_{k_{VRU}}[\,]$ $!=$ *empty AND* $RSU_{k_{OBU}}[\,]$ $!=$ *empty* ***then***
    if (RSU_k *resources sufficient*) *then*
    MEC VRU_C: *Request VRU and OBU context information*
      processing from local $TCC - HINS$
if (*collision predicted event received*) *from local HINS then*
  MEC VRU_C: *Forward alerts to associated VRUs/OBUs*
      else
        proceed;
        end
    else
  MEC VRU_C: *Forward request to Cloud* VRU_C *VNF*;
  Cloud VRU_C *VNF*: *if* (*collision predicted event received*
    from Cloud $TCC - HINS$ *VNF*) *then*
  Cloud VRU_C *VNF*: *Forward alerts*
    to associated MEC VRU_C
      else
        proceed;
      end
    end
else
MEC VRU_C: *Forward request for VRU, OBU context information*
  processing to Cloud VRU_C *VNF*;
  Cloud VRU_C *VNF*: *if* (*collision predicted event received*
    from Cloud $TCC - HINS$ *VNF*) *then*
  Forward alerts to involved MEC VRU_C;
    else
      proceed;
    end
  end
end

identification, that enables efficient processing of massive numbers of trajectories satisfying scalability.

3.2. Trajectory computing component (TCC)

All involved UEs/VRUs and vehicles/OBUs periodically report contextual information to the network; this information is transmitted via a custom message format, in line with the ETSI ITS Cooperative Awareness Message (CAM) format guidelines [42], that includes all the mandatory information from the Basic and the High Frequency (HF) containers described in this message, such as station type, reference position (latitude, longitude, altitude), vehicle heading, speed, vehicle length/width, longitudinal acceleration, etc. TCC processes in real-time this contextual information, received from the OBUs and VRUs and calculates the predicted trajectories. The trajectories are calculated for a time window of T future time slots, i.e. the prediction window. The trajectories are generated by calculating the future position (in the form of latitude, longitude coordinates), along with a set of potential future direction vectors, resulting in a triangular-shaped prediction area (Fig. 3, Fig. 4), formulated by the predicted coordinates and two line segments (i.e., OBU_{a1}, OBU_{a2}, OBU_{b1}, etc.).

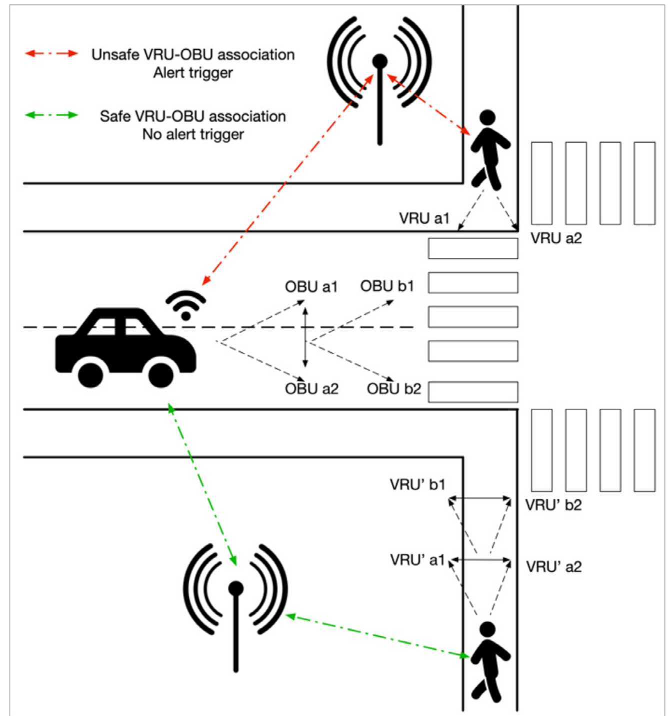


Fig. 3. Modeling future potential direction and location.

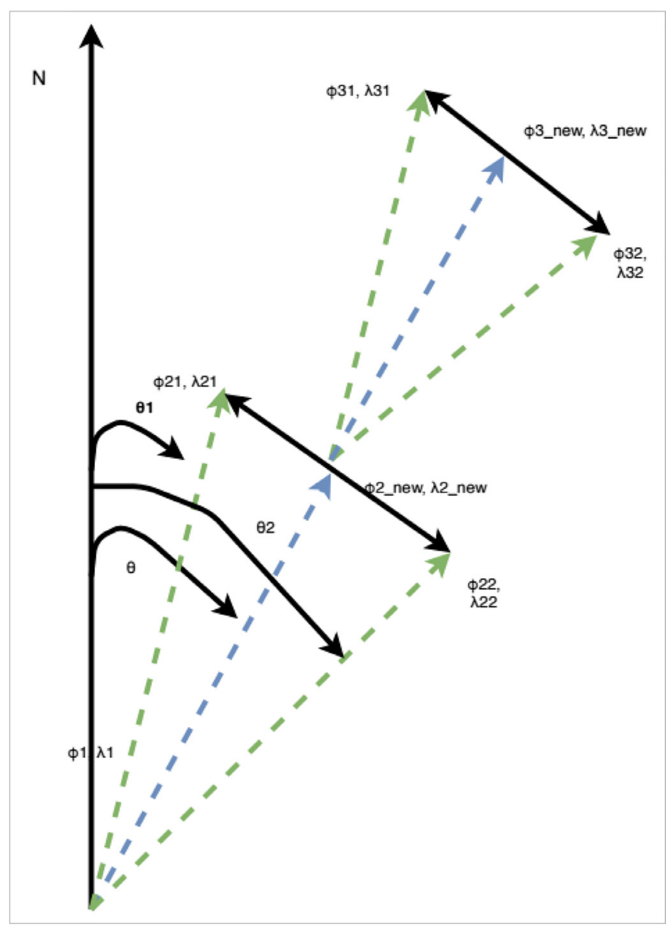


Fig. 4. Modeling future potential direction and location.

Table 1
Predicted positions for one OBU entity (similar tables are constructed for all UEs).

Time	t_0	t_1	t_2	…	t_n	t_{n+1}
$\mathbf{t_0}$	$[\varphi_{OBU\ 0}, \lambda_{OBU\ 0}]$	$r_{OBU\ 1}$	$r_{OBU\ 2}$	…	$r_{OBU\ n}$	$r_{OBU\ n+1}$
$\mathbf{t_1}$		$[\varphi_{OBU\ 1}, \lambda_{OBU\ 1}]$	$r_{OBU\ 2}$	…	$r_{OBU\ n'}$	…
$\mathbf{t_2}$			$[\varphi_{OBU\ 2}, \lambda_{OBU\ 2}]$	…	$r_{OBU\ n''}$	…
…				…	…	…
$\mathbf{t_n}$					…	…

TCC model prediction is portrayed in Fig. 4 in higher detail:

Every UE (OBU/VRU) is modelled as an array of following positions of T columns – depending on the selected time window (see Table 1).

In Table 1 we denote:

$$r = f\left(\varphi_n, \lambda_n, velocity, \theta\right)$$

where φ_n, λ_n, *velocity*, θ, is the latitude, the longitude, the velocity and the constant bearing angle, respectively; *r* is the function that provides the future coordinates as output. It is assumed that the latest position of the UE is known; we calculate the future position for N following time slots; for every time slot of duration t, a calculation of the N future positions is carried out.

We also exploit the Haversine formula [43] in order to predict the future position δ for N time slots:

$$\delta = t * velocity/R,$$

where R is the earth's radius. Additionally:

$$\varphi_2 = arcsin\left(sin\left(\varphi_1\right) * cos(\delta) + cos\left(\varphi_1\right) * sin(\delta) * cos(\theta)\right) and$$
$$\lambda_2 = \lambda_1 + arctan2 * \left(sin(\theta) * sin(\delta) * cos\left(\varphi_1\right), cos(\delta) - -sin\left(\varphi_1\right) * sin\left(\varphi_2\right)\right)$$

In order to increase the flexibility of the mechanism, we employ additional computational methods, able to cope with different types of available context information in different infrastructure set-ups. For example, besides the afore-presented method, the TCC is able to calculate the required trajectories using all or some of the following contextual items: speed, direction, GPS location, acceleration, RSS indicators from neighbor cells, etc. An example follows, with the processing steps, when accelerometer, magnetometer and gyroscope data is available:

- φ_n and λ_n are calculated from a combination of the GPS and the RSSI, every time slot.
- θ is calculated from the fusion of accelerometer, magnetometer and gyroscope data
- Velocity is calculated as $v(t) = \Sigma a^* \delta_t$; for every acceleration sensor value x, y, z:

$$v_x(t) = \sum ax * \delta t$$
$$v_y(t) = \sum ay * \delta t$$
$$v_z(t) = \sum az * \delta t$$
$$v_{total}(t) = \sqrt{v_x^2(t) + v_y^2(t) + v_z^2(t)}$$

The following sub-section focuses on the HINS module, which receives TCC's output predicted trajectories, as input, towards the identification of potential collisions.

3.3. Hazard identification and notification service (HINS)

HINS is responsible for evaluating the computed trajectories, and assessing the possibility of a specific situation to lead to a collision. The main objective of HINS module is to provide an efficient, low time complexity and -as a result- scalable collision identification algorithm, in order to adequately address the computing and time constraints of the automotive domain use cases.

To this end, the main idea behind the identification of a potential collision is that the computed predicted trajectories of all the involved moving entities (OBUs/VRUs) are stored in a common data element, i.e., a multi-dimensional map element comprising the predicted trajectories -in the triangle shape format they were presented in the previous TCC sub-section-, along with the time vector. Thus, the problem of identifying a potential collision between VRU_i and OBU_j among *N* nodes, is broken down into identifying potential intersections of the predicted trajectories -instantly- by the time each single trajectory is computed and added accordingly to the common map/data element. In this manner, traversing all OBUs/VRUs trajectory lists and comparing it in a binary manner is avoided. The advantage of this approach is that the time complexity of the problem is reduced from $O(n^2)$ (in the case each trajectory had to be examined for potential intersection with each one of the rest of the *N-1* moving vehicles' trajectories) to $O(n)$. It should be highlighted that *T* different maps are stored (where *T* is the time window of the previous sub-section), in order to take into account the temporal dimension of each computed trajectory as well.

One of the priorities in HINS design and implementation is to provide the required flexibility, in order to configure the algorithm with varying triggering thresholds for different use cases, nodes' velocity, mobility environments, etc., namely the *Collision threshold*, the *Tolerance window* and the *Immediate threshold*:

a) *Collision threshold:* OBU and VRU moving patterns can be highly dynamic; as a result, several false positive alerts may be generated. Normally, consecutive collision events are generated for – successfully predicted - OBU mobilities, as consecutive time slots trigger a potential hazard. For that purpose, we introduce a threshold that verifies the probability of a collision. Essentially, for each consecutive event that is as evaluated as a possible collision in a specific time frame, a collision counter is incremented, and if this counter exceeds the pre-defined collision threshold, an *Alert* is generated.
b) *Tolerance window:* A potential collision must be identified not only when two predicted trajectories intersect for the same time slot t_n, but also for close -in terms of temporal distance- time slots (i.e., t_{n-d}, t_{n+d}, etc.), as potential prediction errors or changes in the vehicle velocity/direction may occur at any time. The *Tolerance window d* defines how far in time two predicted trajectories intersection should be examined (in other words the difference between time slot t_1 of a trajectory A and t_2 of a trajectory B) in order to increase the respective *collision counter,* defined previously. The *tolerance window* is fine-tuned according to the specific configuration that is expected in a scenario (e.g., urban scenario, highway scenario, etc.), which translates respectively to different average velocities of the involved OBUs and VRUs.
c) *Immediate threshold:* As the position and the bearing angle of the moving nodes change dynamically, there is a possibility, that predicted trajectories that initially intersected, do not intersect for some time (thus, *collision counter* is reset to zero), and then intersect again; this may lead to false negatives. To this end, if a collision is predicted for an immediate event (i.e., in a very short time window from the moment of the prediction), *immediate threshold* is used, ignoring the afore-described *collision counter*. In such a case, an *Immediate Alert* is generated. We set the *immediate threshold* at 3 s, as the minimum time that a driver needs to prevent a collision after getting an alarm is at least 3 s [44,45]. This selection may slightly increase the number of false positives, however more importantly it reduces the false

Table 2
Simulation scenarios.

Scenario	Velocity Range (km/h)	Number of OBUs/VRUs	Scenario Duration (s)
1	(0,20]	4	600
2	(20,50]	8	600
3	(50,100]	10	600

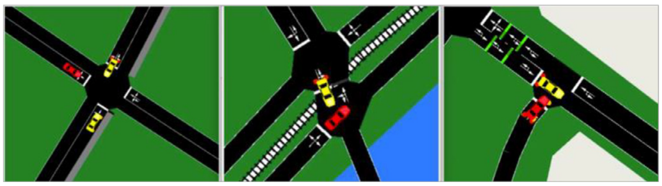


Fig. 5. SUMO simulation scenarios, from left to right: (a) Scenario 1, (b) Scenario 2, (c) Scenario 3.

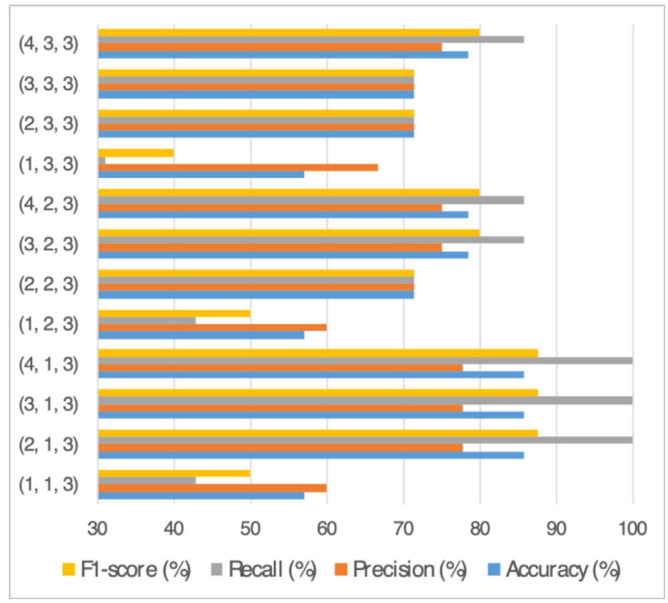


Fig. 6. Algorithm performance for different (tolerance window, collision threshold, immediate threshold) values for Scenario 1.

negatives, i.e., it ensures that OBU and VRU do get a timely notification, in case an imminent collision is about to happen.

The algorithm is described in detail in Algorithm 2.

Based on the TCC and HINS outputs, respective notification messages are generated, and the VRU-C module is responsible to forward the respective messages (*Normal* or *Immediate Alerts*) towards the involved VRUs/OBUs. The respective messages are received by the VRUs in the form of a real-time notification, presented to the user via a mobile application's User Interface (UI); the messages received by the OBUs, are processed by the vehicle running service, towards either automated actions (in the case of fully automated vehicles, such as automated breaking), or presented via the vehicle's UI to the driver. Details with regard to this part, however, are out of scope for this work.

4. Evaluation

This section provides the evaluation outcomes of the proposed framework; in the first part we present the configuration of the collision identification algorithm, via evaluating the F1-score, Recall, Precision and Accuracy metrics for different values of the *tolerance window, collision threshold and immediate threshold* parameters; in the second part of the evaluation, we perform a real-world experiment in different mobility and intersection locations with real, human-controlled vehicles, in order to evaluate the viability of the proposed architecture, the network communication aspects, as well as the validity of the proposed algorithm in a real set-up.

4.1. Algorithm evaluation and fine-tuning

In this first part, our target is to assess the algorithm behaviour for different *tolerance window, collision threshold* and *immediate threshold* parameter values, for different mobility scenarios. We evaluate the algorithm's performance in terms of accuracy, precision, recall and F1-Score parameters and we use the results in order to fine-tune our algorithm for the real network set-up experimentation that is presented in the next sub-section. The metrics assessed relate to the accuracy of predicting actual collisions, false positives, false negatives (missed collisions), etc. It must be noted that the comparison of the proposed algorithm's performance with other existing solutions is out of scope of this experiment. For this round of experiments, we performed several simulations for generating realistic traffic mobility models using SUMO (Simulation of Urban MObility) software [46].

Overall 3 different map environments/scenarios were simulated (Fig. 5) each one of which was run, under 3 different OBUs/VRUs velocity ranges (Table 2); each scenario per velocity range was executed for 10 different runs. That resulted in 90 ($3 \times 3 \times 10$) different simulation executions. Each simulation execution duration was equal to 600 s and the sampling frequency for creating the training and testing datasets for the evaluation of the algorithm was equal to 1 s.

The following graphs (Fig. 6–8) provide the results for different value sets of the [*tolerance window, collision threshold and immediate threshold]* parameters.

The results show that the higher the velocity of the OBUs, the higher the performance of the algorithm (Fig. 7, Fig. 8). This can be explained by the fact that the higher the velocity of a vehicle/VRU is, the less likely it becomes for the moving node to alter its direction, and -as a result- the higher the accuracy of the predicted trajectory is.

Besides identifying the accuracy of the correctly identified collisions, measuring the number of missed collisions is also of utmost importance for the particular use case. As a result, the most crucial KPI, is *Recall,* as it is calculated based both on the true positives, as well as the false negatives of the predictions. We highlight that – particularly for lower velocities (e.g., Fig. 6) - the highest performance, in terms of the *Recall* KPI is observed for the *collision threshold* value set to 1; this translates to the fact that the algorithm exhibits very low false negatives in general. According to this first part of the algorithm evaluation, we configure the parameter set of [*tolerance window, collision threshold, immediate threshold*] to [1,3,4] respectively, as the specific triple of values is one of the sets that exhibits the optimal performance of the algorithm regardless the selected velocity range.

4.2. Real network set-up performance evaluation

The second and main part of the experimentation involved a real network set-up and experimentation environment in the IT-AV Automotive testbed in Aveiro, Portugal [47]. The main objective of this experiment series is to evaluate the viability of the proposed architecture, the network communication aspects (i.e., the end-to-end latency in the

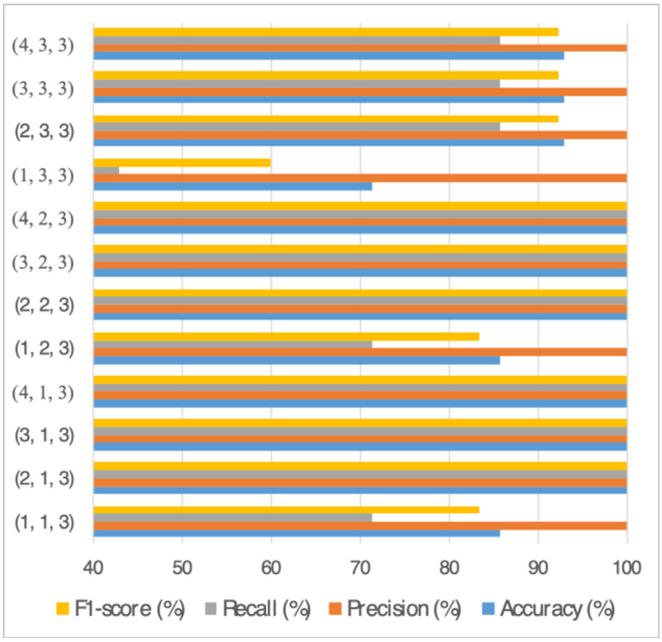


Fig. 7. Algorithm performance for different (tolerance window, collision threshold, immediate threshold) values for Scenario 2.

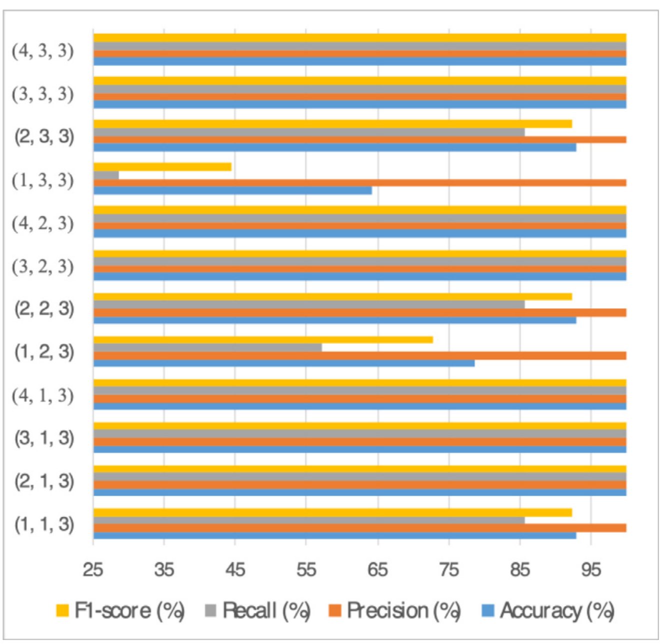


Fig. 8. Algorithm performance for different (tolerance window, collision threshold, immediate threshold) values for Scenario 3.

MEC- and the cloud-based operations), as well as the validity of the proposed algorithm in a real set-up. We juxtapose the performance between the MEC and the Cloud operations, towards identifying the main performance characteristics of the two operations and demonstrate the advantages and disadvantages of the two architectural choices.

The primary goal of the IT-AV testbed (Fig. 9) experimentation was to evaluate the proposed system, based on scenarios, as realistic as possible using real human subjects controlling the OBUs/VRUs, while at the same time to cover a range of diverse parameters, such as the VRUs'/vehicles' distance from the RSU, as well as the velocity and direction patterns of the moving OBU and VRU nodes. The Radio Access

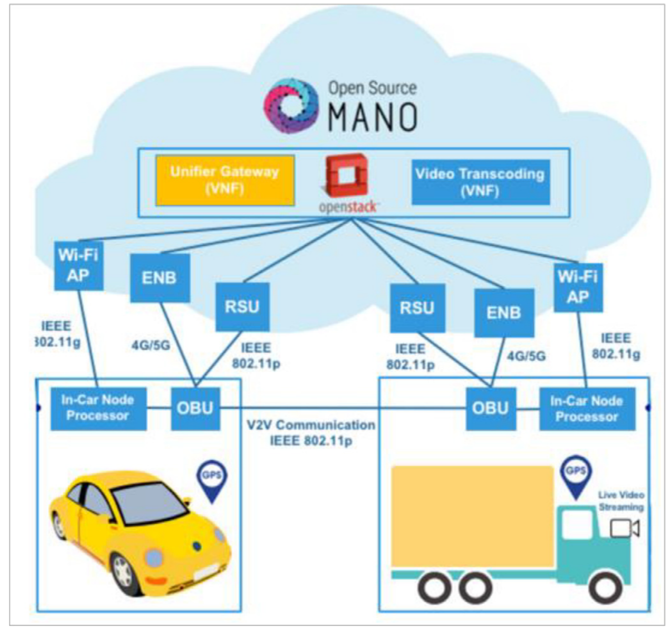


Fig. 9. IT-AV testbed that was used for the evaluation process.

Technology (RAT) that was used for the VRU/OBU – RSU connection was based on the WAVE IEEE 802.11p protocol.

VRU-Safe algorithm's robustness and scalability performance is demonstrated - as it will be shown -, by the fact that it may be applied dynamically for any number of VRUs and OBUs, while also without differentiating between the two types of mobile nodes. For the specific experimentation sessions that we present in this work, we used two OBUs in-car nodes, and one OBU simulating a VRU, communicating over WAVE (802.11p) connection to the RSUs.

4.2.1. Experimentation scenarios overview

Three rounds of experiments were carried out overall (Table 3). The two initial rounds involved real-life mobility scenarios by human-operated connected vehicles in the IT-AV testbed in real-time. The scenarios involved OBU nodes, installed on a vehicle (car), as well as on a bicycle. Three different real-life mobility scenarios took place for Rounds 1 and 2, which attempt to cover different mobility and networking possibilities towards a holistic evaluation of the system. In Scenario 1, both OBUs connect to the same RSU and the processing takes place via the MEC-based operation; in Scenario 2, each OBU connects to a different RSU, and the Cloud VNF-based operation is evaluated. In the final round of experiments (Scenario 3), we focused on the scalability aspect of the proposed scheme. To this end, simulated OBUs' mobility and location information was generated and was transmitted to IT-AV cloud infrastructure, via the respective RSU entity. The reason behind selecting the simulation-based OBU's mobility information in this scenario was the fact that the real testbed's capacity in terms of human subjects and operational OBUs could not address the scenario's requirements; specifically, the number of simulated OBUs increases up to 10 simultaneous OBUs.

Table 3 below provides an overview of the three experimentation scenarios.

Each of the three mobility scenarios is described in further detail.

- Mobility scenario A: In the first scenario, two vehicles are approaching a "T" junction (Fig. 10). The goal is to forward the appropriate notifications to the OBUs at least 10 m before reaching the junction. The vehicles velocity was 10 km/h.
- Mobility scenario B: In the second scenario (Fig. 11), a cyclist follows a non-linear path. Via this mobility path, we target to demonstrate a

Table 3
Rounds of IT-Av testbed scenarios.

Scenario #	Mobility Scenario	RSUs	OBUs/VRUs	Mobility	RAT	Experiment
Infrastructure						
1	A, B, C	1	1/1	Human-operated, real-time	802.11p	IT-Av Testbed OBUs, 1 RSU, 5GINFIRE MANO Cloud infrastructure
2	A, B, C	2	1/1	Human-operated, real-time	802.11p	IT-Av Testbed OBUs, 2 RSUs, 5GINFIRE MANO Cloud infrastructure
3	SUMO-based mobility	1	10/0	Simulated mobility applied on real-time network operation	802.11p	5GINFIRE MANO Cloud infrastructure, 1 IT-Av Testbed RSU

Algorithm 2 TCC and HINS algorithmic steps.

Variables $i,\ j,\ t, c_{ij} \in N$;
Parameters: $M, N, T, T_{window}, T_{immediate}, c_{threshold} \in N$;
Initialize pointers for M VRUs, N OBUs: $i \leftarrow 0, j \leftarrow 0$;
while $i < M$ *do*
  $Initialize\ t = 0$;
  while $t < T$ *do*
    VRU_C: *Receive context information from* VRU_i;
TCC: *Compute predicted trajectories for* VRU_{it} *for T next timeslots*;
      $t++$;
    end
    $i++$;
end
while $j < N$ *do*
  $Initialize\ t = 0$;
  while $t < T$ *do*
  VRU_C: *Receive context information from* OBU_j;
TCC: *Compute predicted trajectories for* OBU_{jt} *for T next timeslots*;
    $t++$;
  end
  $j++$;
end
Reset pointers for trajectory collision identification: $i \leftarrow 0, j \leftarrow 0, t \leftarrow 0$;
Initialize collision counter: $c_{ij} \leftarrow 0$;
for each $i < N, j < M$ *do*
  if $t < T$ *then*
HINS: *if*(*Collision is identified between* VRU_i *and* OBU_j)*AND*($t < T_{window}$) *then*
  $Increment\ counter\ for\ VRU_i\ and\ OBU_j:\ c_{ij}++$;
    if $t < T_{immediate}$ *then*
      trigger Immediate Alert;
      $c_{ij} \leftarrow 0$;
    $else\ if\ (t < T\ AND\ c_{ij} == c_{threshold})\ then$
      trigger Normal Alert;
      $c_{ij} \leftarrow 0$;
    $else\ proceed\ to\ next\ time\ slot:\ t++$;
  end
 end
end

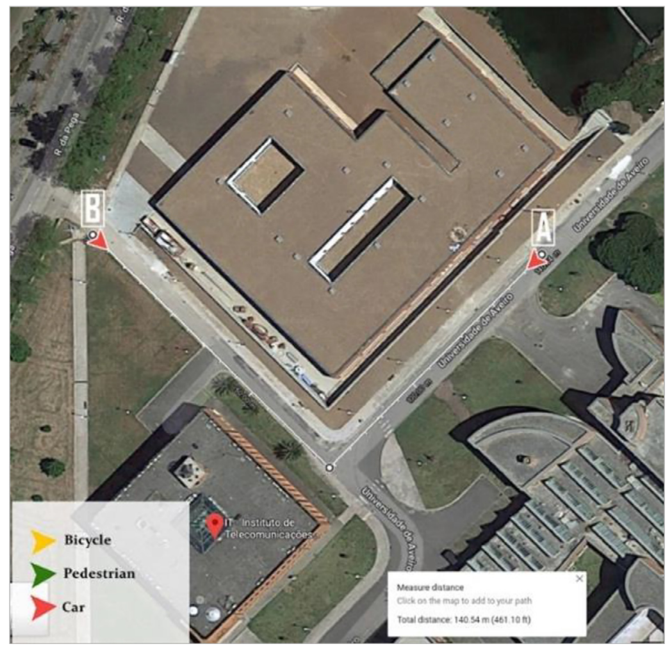


Fig. 10. Overview of the IT-AV testbed scenarios (Scenario A).

more challenging scenario with a variety of manoeuvres and changes of direction. The bicycle speed is 6 km/h. From the other side of the junction, there is a vehicle approaching at 10 km/h.

- Mobility scenario C: In the last mobility scenario (Fig. 12), a vehicle (A) is approaching a parking slot area; at the same time, a second vehicle (B) is about to depart from the parking slot area. This scenario has been selected, as very often, accidents take place in real-life in the specific road, due to other, parked vehicles that limit the visibility of the drivers exiting the parking slot area.

4.2.2. Results

In all afore-presented experiments, the primary objective is to trigger potential collision situations between the mobile nodes/OBUs and VRUs, and assess the responsiveness and performance of the system in an end-to-end manner.

In this evaluation round, we focus on the critical end-to-end latency KPI, which is defined as the overall time elapsed, from the time that the system receives a specific OBU-oriented contextual information (that eventually triggers a "potential imminent collision" alert), until the time, that all involved OBUs/VRUs receive the actual notification from the system (i.e., the HINS module in particular). The end-to-end latency is calculated as the sum of (a) the overall communication latency (radio propagation, fronthaul-, backhaul-links' related latency), (b) the control plane induced delay (i.e., VRU-C control processing-related delay) and (c) the overall processing time of the computing modules (MEC- or cloud-based VNFs). Before the experiments, all VRUs and OBUs clocks were synchronized at the order of 10^{-3s}, so as to satisfy the KPI assessment that followed on the level of milliseconds.

4.2.2.1. Experiment 1. In the first round of experiments, the VRUs and OBUs are connected to the network via the same RSU node; the system is thus capable to instantly process the mobility-related, contextual information both directly, in a MEC-based, as well as in a cloud-based mode: we juxtapose the performance of the system in the two modes, attempting to assess the tradeoff between the reduced communication latency in the MEC operation, with the minimized computational delay in the cloud-based operation (due to the cloud resources superiority when comparing to the limited, MEC computing resources). The results, which follow for the different mobility scenarios, demonstrate those assessment outcomes.

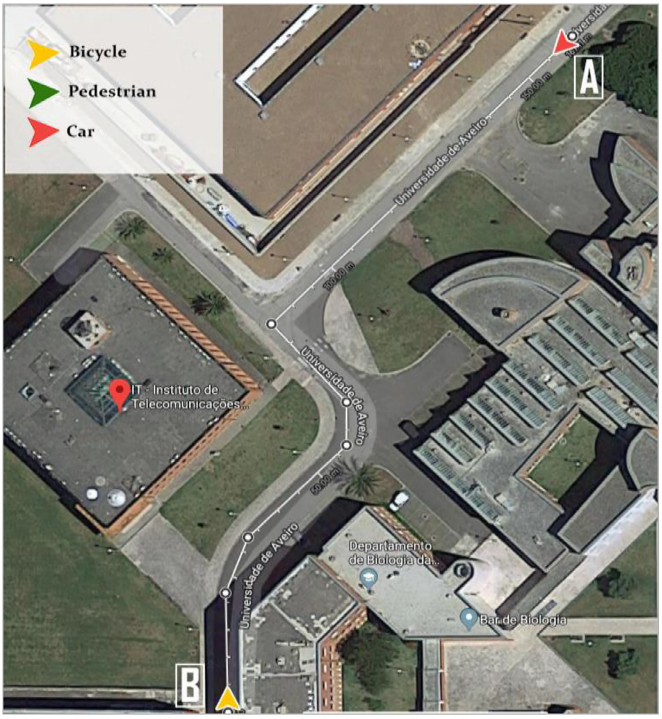


Fig. 11. Overview of the IT-AV testbed scenarios (Scenario B).

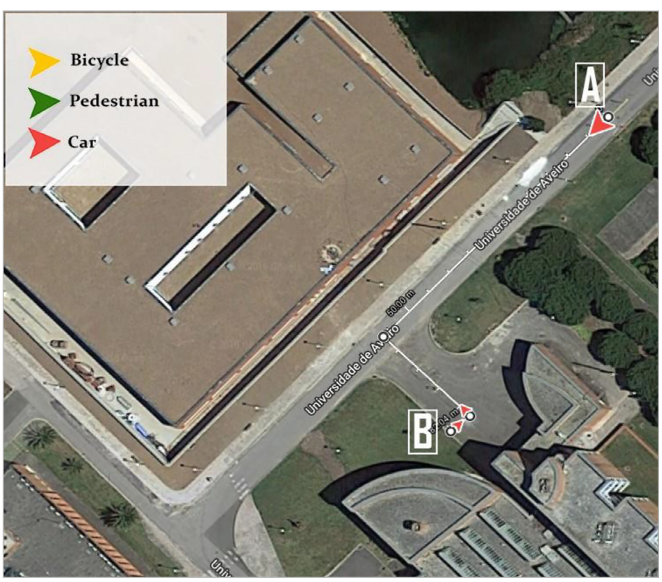
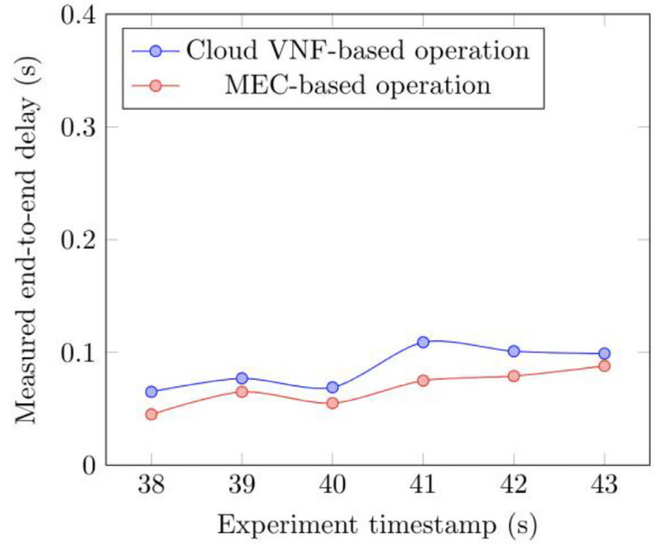
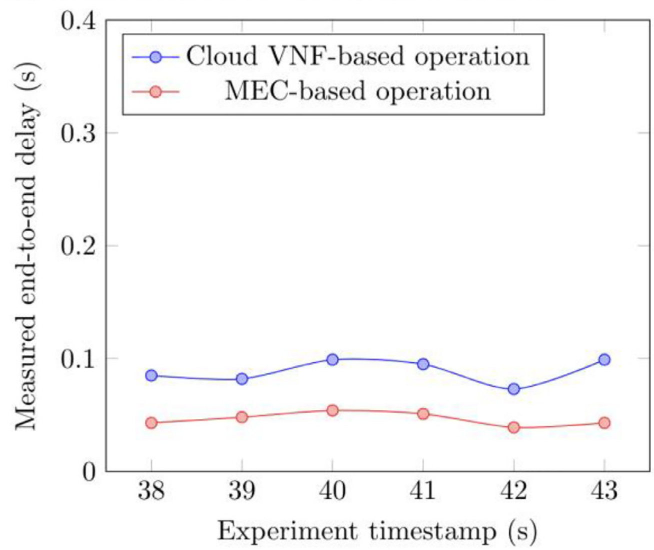


Fig. 12. Overview of the IT-AV testbed scenarios (Scenario C).



(a) Measured delay for OBU_1 in Scenario 1A



(b) Measured delay for OBU_2 in Scenario 1A

Fig. 13. End-to-end delay for Scenario 1A.

Scenario 1A

The first scenario involves two cars (OBU_1 and OBU_2 respectively) approaching a T-junction. The notification for a potential collision is triggered before the OBUs reach the junction. The cars started at the same distance from the crossroad and their velocity was 10 km/h. Fig. 13 illustrates the measured end-to-end delay for the respective alert message to be delivered to each one of the two OBUs; the values that are illustrated on the horizontal axis represent specific timestamps of the experiment, for which a collision event was generated. The sampling rate of the experiment was set to 60 Hz; as a result, each timestamp corresponds to one elapsed second.

As it can be inferred from the two plot graphs for the two OBUs, the MEC-based performance outperforms the cloud VNF-based one; for OBU_1 the MEC-based operation illustrates a mean (resulting from the 6 different values) end-to-end delay of 67.83 ms, while the cloud-based operation has demonstrated a mean delay of 86.66 ms. The results are similar for OBU_2. The difference is explained by the fact that in the MEC-based operation the processing for the two OBUs context takes place locally using MEC resources, while in the VNF-based processing, the information must be transmitted via more network hops to the cloud, aggregating higher end-to-end delay. Additionally, the processing load resulting from the number of two OBUs can be adequately addressed by the MEC-based resources.

It should be highlighted that by investigating further the aggregated delay portions, it results that the processing delay (end-to-end delay minus the propagation/forwarding delay minus the VRU-C/control plane induced delay) of the cloud-based operation is slightly lower, due to superior, centralized computing resources. Nevertheless, although there is a considerable difference between the two operations' end-to-end performance, both showcase acceptable KPI results: the requirement of the particular application is to forward a prediction alert to the involved OBUs/VRUs earlier - in the order of hundreds of milliseconds - before a

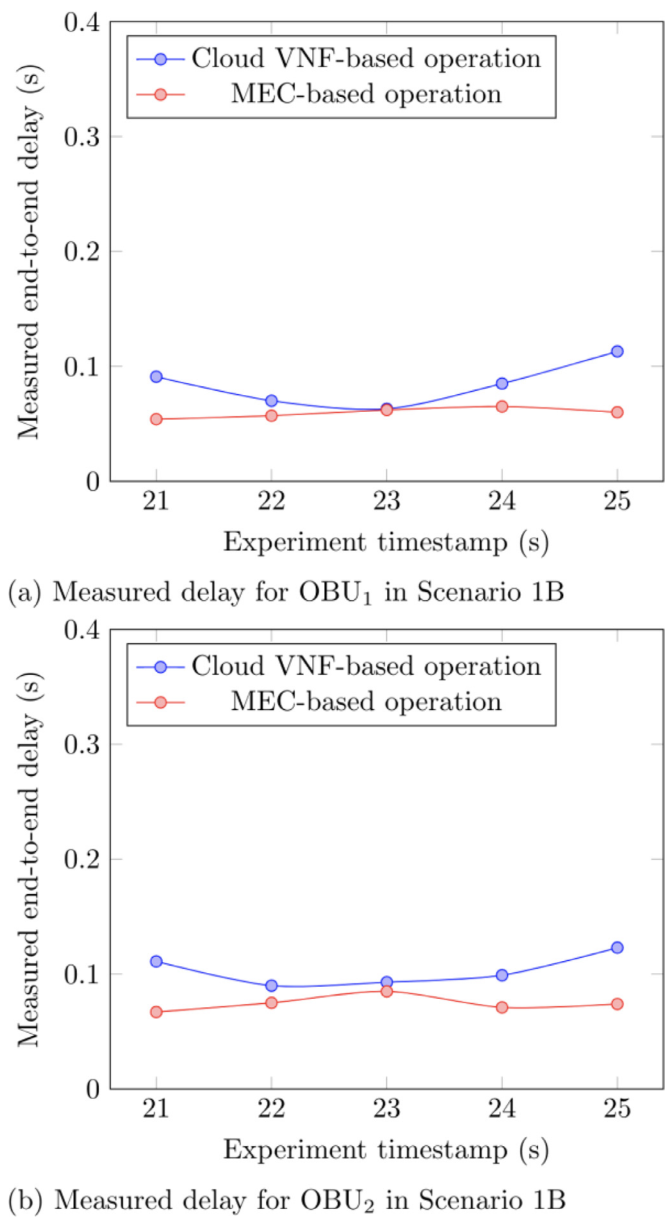


Fig. 14. End-to-end delay for OBU_1 in Scenario 1B

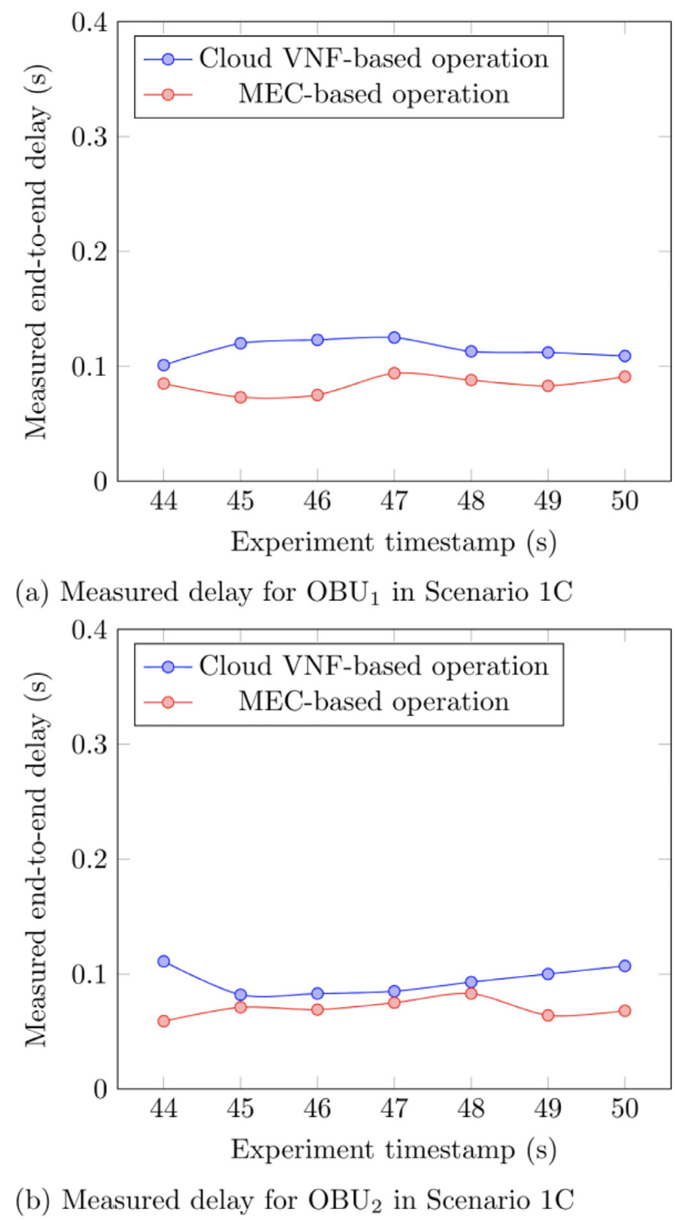


Fig. 15. End-to-end delay in Scenario 1C.

potential predicted collision is about to occur; the order of the end-to-end delay in this experiment is in the order of few tens of milliseconds, thus it is considered to satisfy the required service.

Scenario 1B

As already presented, in scenario 1B there is a cyclist that follows a non-linear path (Fig. 11). Similarly, with the previous scenario, Fig. 14 illustrates the measured end-to-end delay for the alert message to be delivered to the involved OBU entities.

Firstly, the collision identification service managed to successfully identify the hazardous situation, despite the challenging mobility pattern of the cyclist. Additionally, the results of Scenario 1B are similar, in the sense that the cloud-based operation once more demonstrates in general a higher end-to-end delay (84.4 ms and 103.2 ms for OBU_1 and OBU_2 respectively) comparing to the MEC-based operation (59.6 ms and 74.4 ms).

Scenario 1C

Fig. 15(a) and (b) illustrate the measured end-to-end delay for Scenario 1C. Once more, the superiority of the MEC-based operation in this round of scenarios is confirmed by the end-to-end delay results.

4.2.2.2. Experiment 2. The second round of experiments, involves an identical round of mobility scenarios (Figs. 10–12); in this particular set of experiments, however, a second RSU node is also deployed, resulting in 2 RSU nodes serving 2 OBUs (one RSU for each OBU respectively). This approach is followed in order to validate the system's performance when no MEC-based operation takes place; all the trajectory prediction, as well as hazard identification processes (TCC and HINS modules' operation respectively) takes place on the IT-AV testbed cloud. The TCC and HINS operations, in the form of VNFs receive contextual information from the moving OBUs, via the respective associated RSUs, and forward the respective alert events via the same network paths.

Scenario 2A

This is the replication of Experiment's 1 scenario A (Fig. 10) where two cars are approaching a T-junction, adapted to the Experiment 2 set-up. OBU_1 is connected to RSU_1 and OBU_2 is connected to RSU_2 respectively. The results of the experiment are illustrated in Fig. 16:

As it can be inferred by the two figures, both OBUs present a similar performance: OBU_1 –RSU_1 demonstrate an average end-to-end delay of 74.62 ms and OBU_2-RSU_2 demonstrate a slightly lower measured

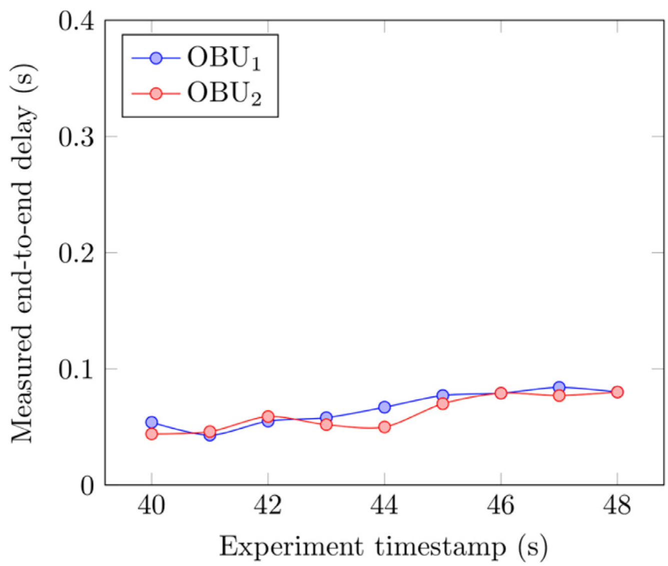


Fig. 16. End-to-end delay in Cloud-based operation for Scenario 2A.

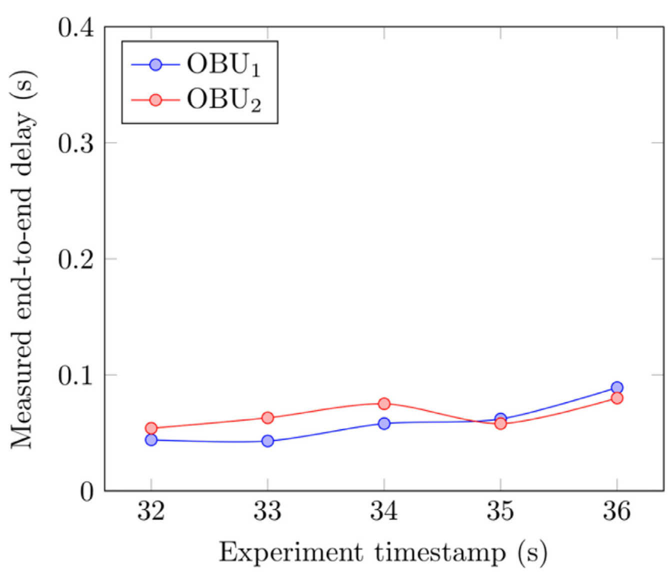


Fig. 17. End-to-end delay in Cloud-based operation for Scenario 2B.

end-to-end delay of 69.60 ms. One significant deduction is the robust performance of the cloud-based operation, which demonstrates a stable performance for the two OBU- RSU, associations; at the same time, it is interesting to compare those delay results with the cloud-based operation performance of Scenario 1A (i.e., the identical scenario with 1 RSU). In Scenario 1B, MEC VRU-C directly forwards the context information received from the two sources towards the cloud for the TCC and HINS VNF execution, as it correctly identifies that the list of OBUs/VRUs is not sufficient for collision identification; at the same time, MEC VRU-C does not have to handle any MEC-based processing context besides the cloud-based one; this results in a slight end-to-end delay reduction, comparing to Scenario 1A, taking also into account that the processing time in the cloud is identical in both scenarios.

Scenario 2B

The identical scenario of 1.B – but this time with 2 RSUs - involves one bicycle (in the role of the VRU) and one vehicle, both approaching at the T-junction. Fig. 17 illustrates the results of the scenario.

Similarly, with Scenario 2A, the cloud-based operation mode demonstrates a similar performance for both OBU/RSU associations, well below 100 ms, as well improved comparing with the identical scenario 1B with 1 RSU.

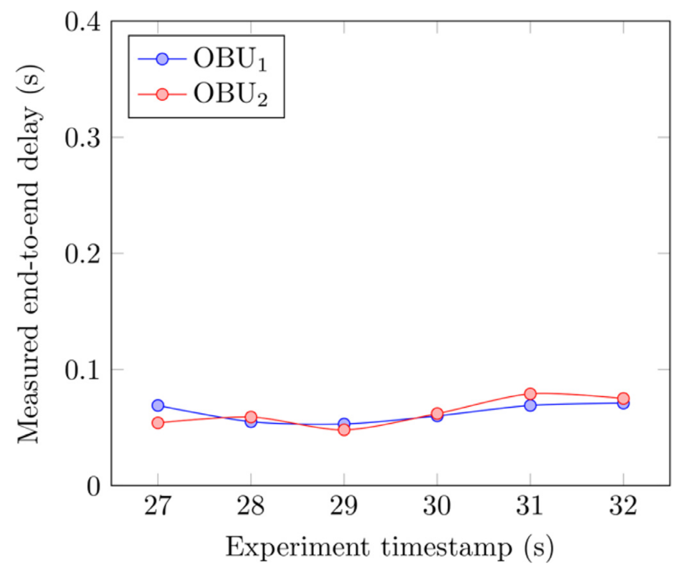


Fig. 18. End-to-end delay in Cloud-based operation for Scenario 2C.

Scenario 2C

The final scenario of the second round repeats the “limited visibility when exiting a parking area” scenario (Fig. 12) on cloud-based operation only. The results are illustrated in Fig. 18 that follows:

Once more, the performance shows a clear improvement over the respective Scenario 1C and practically identical results between the two OBU-RSU associations, confirming the cloud operation stable performance.

4.2.2.3. Experiment 3. The main objective of Scenario 3 is to validate the system in terms of scalability performance. This focuses on assessing the overall performance of the system at an increasing number of mobile OBUs/VRUs that must be served. Due to limitations imposed by the testbed’s resources in terms of availability of OBUs and vehicles, the mobile OBUs have been modelled based on the SUMO simulation software, which provides simulated mobility data of OBUs into the real MEC- and cloud- facilities in the IT-AV testbed for further processing.

Fig. 19 illustrates the overall number of identified anticipated collision events for increasing number of OBUs (ranging from two OBUs up to ten) for both Cloud- and MEC-based operation.

As it was expected the results are identical for the two operation modes (both MEC- and Cloud-based), as the algorithms of the collision identification are the same in both, MEC and cloud deployments; for the same SUMO mobility input datasets, the same output (identified collisions) was received. The main insight from this outcome is that the number of identified collisions for an increasing number of roaming OBUs increases at an exponential rate (3 collisions alerts are generated for 2 OBUs, while 44 collisions are identified in the 10 OBUs case); this is explained by the fact that we selected the simulated mobility patterns not to follow any geographical constraints (e.g. road limits); the simulated vehicles are moving at random directions within a pre-defined rectangular area. This resulted in a highly complex mobility environment with an ever-increasing number of collision identifications, towards the evaluation of the scalability of the system.

The following figure (Fig. 20) illustrates the end-to-end delay for the alert message delivery, for the two operation modes, for the same OBU mobility patterns.

The results in this case are completely different, demonstrating a vast superiority of the cloud-based operation for increasing numbers of OBUs. It is crucial to highlight that all OBUs’ are managed by one

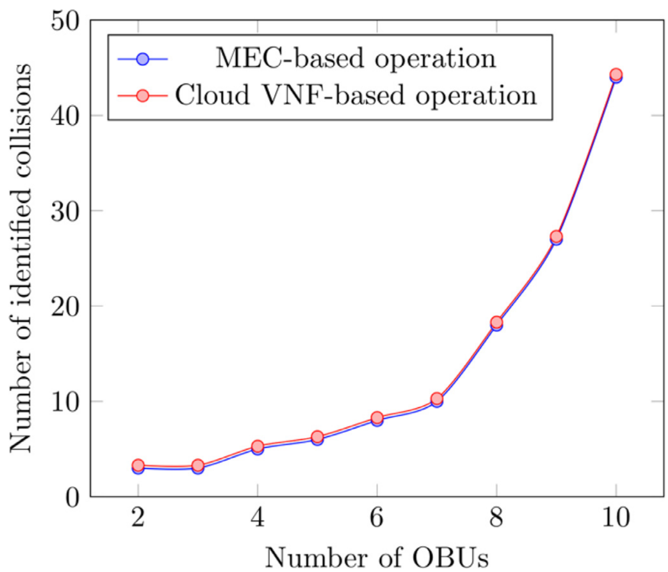


Fig. 19. MEC- and Cloud-based operation's identified collision events for increasing number of OBUs.

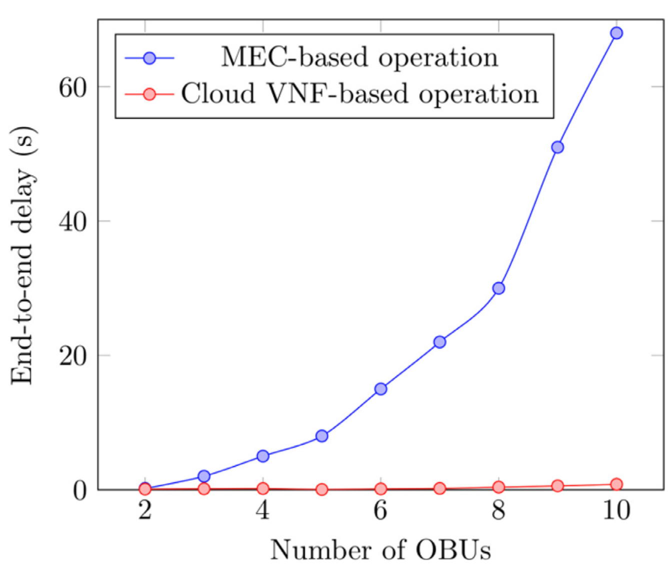


Fig. 20. Comparison between the RSU-based and the VNF-based operations for increasing number of OBUs.

RSU. The computational overhead imposed by the specific infrastructure (RSU limited computing capabilities) practically makes the specific performance unacceptable for such a V2X service with such critical requirements in terms of end-to-end delay. On the contrary, in the cloud-based operation, an almost linear increase is measured -reaching a highest value of 0.8 s for processing simultaneously mobility information from 10 OBUs-.

5. Conclusion and future directions

A hybrid architecture for V2X services was presented, which exploits MEC and cloud-based resources in a coordinated manner towards optimizing the end-to-end performance of the system. On top of the proposed architecture a novel V2X service - namely VRU-safe - was presented and thoroughly evaluated in a real network wireless automotive testbed, and in a diverse set of mobility scenarios. All in all, it is shown that VRU-safe service is able to provide accurate and timely predictions of potential imminent collision events and take advantage of the distributed nature of the VRU-Safe system operation.

The experiments showcased both the MEC-based VRU-Safe system's, as well as the cloud VNF-based operation performances, in terms of end-to-end delay, number of collision events detected and system scalability. Both operations always correctly identified the potential hazardous situations and timely forwarded the respective notifications to the VRUs and OBUs. The MEC operation demonstrated a superior performance in certain scenarios with a small number of OBUs, while the cloud-based operation outperformed the MEC operation in upscaled scenarios and showed overall a robust and stable performance. This leads us to infer that if MEC computing resources are high enough for being able to manage higher number of OBUs, a dynamic resource selection approach depending on real-time network resources availability information can potentially lead to high gains. It is important to highlight that although the end-to-end latency requirements for V2P scenarios, such as the VRU safety use case are satisfied with the presented evaluation results, such as a system will need to be further optimized in terms of this KPI, for automated driving scenarios, in order to achieve end-to-end latencies below the threshold of 20 ms.

To this end, one of the primary next steps for the proposed system is further improve the efficiency of the algorithm in order to reduce the end-to-end latency, as well as perform further experimentation using enhanced MEC computing resources, in order to overcome the related processing limitations for upscaled scenarios and -thus- fully exploit the potential of the flexibility of the system in diverse contexts and situations. It is of utmost importance as well, to further improve the performance of the collision identification service, i.e. both via increasing its collision identification accuracy, as well as via reducing the false negatives and the false positives, and specifically for the low velocity ranges of the moving vehicles/VRUs. This will be realized by federating the existing trajectory prediction mechanism by deep learning-based prediction techniques, such as Long Short Term Memory (LSTM) network-based prediction.

Declaration of Competing Interest

None.

Acknowledgments

The research leading to these results has been performed under the H2020 project 5GinFIRE [48].

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.comnet.2020.107150.

References

[1] M. Simsek, D. Zhang, D. Ohmann, M. Matthe, G. Fettweis, On the Flexibility and Autonomy of 5G Wireless Networks, IEEE Access, 2017, doi:10.1109/ACCESS.2017.2715318.
[2] 5G Automotive Association (5GAA), Toward fully connected vehicles: Edge computing for advanced automotive communications. December 2017.
[3] 5G Infrastructure Public Private Partnership (5G-PPP), 5G Strategic deployment agenda for connected and automated mobility in Europe, October 2019. https://5g-ppp.eu/wp-content/uploads/2019/10/20191031-Initial-Proposal-5G-SDA-for-CAM-in-Europe.pdf [Accessed November 2019].
[4] J. Lee, Y. Kim, Y. Kwak, et al., LTE-Advanced in 3GPP Rel -13/14: An evolution toward 5G, IEEE Communications Magazine, 2016, doi:10.1109/MCOM.2016.7432169.
[5] S. Chen, J. Hu, Shi, et al., Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G, IEEE Communications Standards Magazine, 2017, doi:10.1109/MCOMSTD.2017.1700015.
[6] 5G communication automotive research and innovation (5G Car) H2020 project, https://5gcar.eu/ [Accessed November 2019].

[7] 5G communication automotive research and innovation (5G Croco) H2020 project, https://5gcroco.eu/ [Accessed November 2019].
[8] ETSI - Multi-access Edge Computing (MEC), Study on MEC Support for V2X Use Cases, Sept. 2018.
[9] 5GAA, Toward Fully Connected Vehicles: Edge Computing White Paper, 2017 5GAA Automotive Association.
[10] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, W. Xu, Connected roads of the future: Use cases, requirements, and design considerations for vehicle-To-everything communications, IEEE Veh. Technol. Mag. (2018).
[11] B.M. Masini, A. Bazzi, E. Natalizio, Radio access for future 5G vehicular networks, 86th IEEE Vehicular Technology Conference, 2017.
[12] Toward fully connected vehicles: edge computing for advanced automotive communications, White Paper. Available at http://5gaa.org/wp-content/uploads/2017/12/5GAA_T-170219-whitepaper-EdgeComputing_5GAA.pdf [Accessed November 2019].
[13] 3GPP TR 23.786 v0.8.0. Technical specification group services and system aspects; Study on architecture enhancements for EPS and 5G system to support advanced V2X services, Rel. 16, Aug. 2018.
[14] C. Campolo, A. Molinaro, A. Iera, F. Menichella, 5G network slicing for vehicle-to everything services, IEEE Wireless Commun. 24 (6) (2017) 38–4.
[15] S. Chen, J. Hu, Shi, et al., Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G, IEEE Communications Standards Magazine, 2017, doi:10.1109/MCOMSTD.2017.1700015.
[16] F. Giust, V. Sciancalepore, D. Sabella, et al., in: Multi-Access Edge Computing: The Driver Behind the Wheel of 5G-Connected Cars, IEEE Communications Standards Magazine, 2018, p. 2, doi:10.1109/MCOMSTD.2018.1800013.
[17] A Platform for Computing at the Mobile Edge: Joint Solution with HPE, Saguna, and AWS, Amazon White Paper, Available at https://d1.awsstatic.com/whitepapers/platform-for-computing-at-the-mobile-edge-hpe-saguna-aws.pdf.
[18] D. Grewe, M. Wagner, M. Arumaithurai, et al., Information-centric mobile edge computing for connected vehicle environments: challenges and research directions, in: Proceedings of the 2017 Workshop on Mobile Edge Communications (Mecomm '17), 2017, doi:10.1145/3098208.3098210.
[19] P. Corcoran, S.K. Datta, Mobile-Edge Computing and the Internet of Things for Consumers: Extending Cloud Computing and Services to the Edge of the Network, IEEE Consumer Electronics Magazine, 2016, doi:10.1109/MCE.2016.2590099.
[20] K. Zhang, Y. Mao, S. Leng, et al., Mobile-Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Off-Loading, IEEE Vehicular Technology Magazine, 2017, doi:10.1109/mvt.2017.2668838.
[21] Emara, M., Filippou, C. M., Dario, S. (2017). MEC-aware Cell Association for 5G Heterogeneous Networks.
[22] A. Aissioui, A. Ksentini, A.M. Gueroui, T. Taleb, On Enabling 5G Automotive Systems Using Follow Me Edge-Cloud Concept, IEEE Transactions on Vehicular Technology, 2018, doi:10.1109/TVT.2018.2805369.
[23] T. Taleb, S. Dutta, A. Ksentini, et al., Mobile Edge Computing Potential in Making Cities Smarter, IEEE Communications Magazine, 2017, doi:10.1109/MCOM.2017.1600249CM.
[24] L. Li, M. Siew, T.Q.S. Quek, Learning-based pricing for privacy-preserving job offloading in mobile edge computing, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 4784–4788, doi:10.1109/ICASSP.2019.8682862.
[25] P. Mach, Z. Becvar, Mobile Edge Computing: A Survey on Architecture and Computation Offloading, IEEE Communications Surveys and Tutorials, 2017, doi:10.1109/COMST.2017.2682318.
[26] M. Malinverno, G. Avino, C.E.E. Casseti, et al., Edge-based Collision Avoidance for Vehicles and Vulnerable Users, IEEE Vehicular Technol. Mag. (2019), doi:10.1109/MVT.2019.2953770.
[27] A. Napolitano, et al., Implementation of a MEC-based vulnerable road user warning system, 2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive, 2019.
[28] A. Jahn, K. David, S. Engel, 5G / LTE based protection of vulnerable road users: detection of crossing a curb, 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall), 2015, doi:10.1109/VTCFall.2015.7390782.
[29] M. Emara, M.C. Filippou, D. Sabella, MEC-Assisted End-to-End Latency Evaluations for C-V2X Communications, 2018 European Conference on Networks and Communications, EuCNC 2018, 2018, doi:10.1109/EuCNC.2018.8442825.
[30] T. Zhang, Data Offloading in Mobile Edge Computing: A Coalition and Pricing Based Approach, IEEE Access, 2017, doi:10.1109/ACCESS.2017.2785265.
[31] C.-R. Storck, D-F. Fátima, A 5G V2X Ecosystem Providing Internet of Vehicles, Sensors (Basel, Switzerland) 19 (3) (2019) 550 29 Jan., doi:10.3390/s19030550.
[32] A. Sallab, M. Abdou, E. Perot, S. Yogamani, Deep reinforcement learning framework for autonomous driving, Electronic Imaging. 2017 (2017) 70–76, doi:10.2352/ISSN.2470-1173.2017.19.AVM-023.
[33] H. Ye, G.Y. Li, B.H.F. Juang, Deep Reinforcement Learning Based Resource Allocation for V2V Communications, IEEE Trans. Vehicular Technology (2019), doi:10.1109/TVT.2019.2897134.
[34] J. Kotte, C. Schmeichel, A. Zlocki, et al., Concept of an enhanced V2X pedestrian collision avoidance system with a cost function–based pedestrian model, Traffic Injury Prevent. (2017), doi:10.1080/15389588.2017.1310380.
[35] J. Wang, Y. Shao, Y. Ge, Y. Rundong, A survey of vehicle to everything (V2X) testing, Sensors 19 (2019) 334, doi:10.3390/s19020334.
[36] B. Wulfe, S. Chintakindi, S.T. Choi, et al., Real-time Prediction of Intermediate-Horizon Automotive Collision Risk, AAMAS, 2018.
[37] M. Rondinone, T. Walter, R. Blokpoel, J. Schindler, V2X communications for infrastructure-assisted automated driving, 19th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2018, 2018, doi:10.1109/WoWMoM.2018.8449775.
[38] B. Blanco, J.O. Fajardo, I. Giannoulakis, et al., Technology Pillars in the Architecture of Future 5G Mobile Networks: NFV, MEC and SDN. Computer Standards and Interfaces, 2017, doi:10.1016/j.csi.2016.12.007.
[39] A. Aissioui, A. Ksentini, A.M. Gueroui, T. Taleb, On Enabling 5G Automotive Systems Using Follow Me Edge-Cloud Concept, IEEE Transactions on Vehicular Technology, 2018, doi:10.1109/TVT.2018.28053691.
[40] L. Liang, H. Ye, G.Y. Li, Toward intelligent vehicular networks: a machine learning framework, IEEE IOT J. (2019), doi:10.1109/JIOT.2018.2872122.
[41] A. Bazzi, C. Campolo, B.M. Masini, et al., . Enhancing cooperative driving in IEEE 802.11 vehicular networks through full-duplex radios, IEEE Trans. Wireless Commun. (2018).
[42] ETSI EN 302 637-2. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service.
[43] Haversine formula, https://en.wikipedia.org/wiki/Haversine_formula [Accessed November 2019]
[44] X. Yan, Y. Zhang, L. Ma, The influence of in-vehicle speech warning timing on drivers' collision avoidance performance at signalized intersections, Transp. Res. Part C (2015), doi:10.1016/j.trc.2014.12.003.
[45] N. Lubbe, E Rosen, Pedestrian crossing situations: Quantification of comfort boundaries to guide intervention timing, Accident Anal. Prevent. 71 (2014) 261–266.
[46] SUMO, Simulation of Urban Mobility, http://sumo.sourceforge.net [Accessed November 2019].
[47] IT-AV Automotive Testbed, Aveiro, Portugal, https://5ginfire.eu/it-av-automotive-testbed/ [Accessed November 2019].
[48] H2020 5GinFIRE, http://wiki.5ginfire.eu/ [Accessed November 2019]

Dr. Sokratis N. Barmpounakis is a Postdoctoral Research Associate in SCAN group, Dept. of Informatics and Telecommunications, in the National and Kapodistrian University of Athens (NKUA). He holds a Ph.D. in "Context-based Resource Management and Slicing for SDN-enabled 5G Smart, Connected Environments", since May 2018. Dr. Barmpounakis obtained his Engineering Diploma in Electrical and Computer Engineering, from the National Technical University of Athens (NTUA) in 2010. After completing his studies, he moved to Geneva, Switzerland. At University of Geneva, he worked for 2 years as a Researcher/Developer in various projects, Swiss and European, as a member of the QoL group. Since March 2013, he is a researcher in the SCAN group, actively involved in numerous European Projects and industrial contracts, having also undertaken managerial roles, coordinating the University of Athens team. Dr. Barmpounakis is a co- inventor of 2 patents, approved and filed in the World Intellectual Property organisation. His main fields of interest are: 5G, LTE, Software Defined Networking, Context-Aware Resource and Mobility Management and Collaborative Data Sharing mechanisms. He is also a member of the Technical Chamber of Greece.

Mr. George Tsiatsios obtained his four-year Bachelor of Science (B.Sc.) from National and Kapodistrian University of Athens (NKUA), in the Department of Informatics and Telecommunications, with a specialization in Telecommunications and Networks. He is currently a student at NCSR Demokritos, in the field of Data Science and works as a Researcher at the Software Centric & Autonomic Networking (SCAN) Lab of UoA. His main fields of interest are AI, Data Analytics, Software-Defined Networking (SDN), Software-Defined Wireless Local Area Networking (SDWLAN), Future networks and 5G.

Mr. Michael Papadakis graduated from the Department of Informatics and Telecommunications, NKUA, and is currently completing his M.Sc. at the University of Piraeus, Dept. of Digital Systems, Object Oriented Architecture and Services. He is collaborator with SCAN group, University of Athens and with TERRA SPATIUM, Geo-information and Space Products & Services S.A. His field of studies include Cloud Computing, SDN, SDR, 5G communications, as well as conceptual models, system analysis, decentralized architectures and smart maritime applications. He is currently researching on computer vision and deep neural networks on real time communications.

Mr. Evangelos Mitsianis holds a Computer Engineer degree from University of Thessaly and a Msc. degree in Machine Learning from National and Kapodistrian University of Athens. Currently, he is Ph.D. candidate of National and Kapodistrian University of Athens in Informatics and Telecommunications department. He serves as research associate in Self-evolving Cognitive and Autonomic Networking Lab (Scan-Lab). Mr. Mitsianis main research interests are focus in designing reliable, fast and power-aware artificial intelligence and bigdata applications, using cutting edge technologies. Specifically, he is interested in algorithms and computational methods used in Data Science, Machine learning and Deep Learning for Image Analysis, Audio-Speech recognition, and Video scene recognition. Also, he focus on applied artificial intelligence for next generation telecommunication networks.

Mr. Nikolaos Koursioumpas holds a M.Sc. degree with a specialization in Computer Networking at the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens. He obtained his B.Sc from the same University with a specialization in Networks and Telecommunications. He is currently working as a research associate in Big Data Engineering in SCAN lab.

Assoc. Prof. Nancy Alonistioti is an Associate Professor in the Department of Informatics Telecommunications (Dept. of Informatics and Telecommunications, NKUA). She has over 20 years of experience in numerous national and European projects, including project/ technical management experience. She is currently leading the SCAN group activities in the Dept. of Informatics in the National, NKUA. She has served as member of the Future Internet Assembly Steering Committee. She has also undertaken management activities as a PMT member and WP Leader of the FP6 IST- E2R and FP7 E3 projects and the PC of FP7 ICT Self-NET and FP7 ICT CONSERN projects. She is member of the ETSI Experts group and the Greek standardization group ELOT (5G, smart citiy autonomic communications). She has over 100 publications in the area of mobile networks, Future Internet/NGI, cognitive management, autonomic communications and reconfigurable mobile systems. She is co-author of 4 WO Patents and has more than 2500 citation in her work.