International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 11 (2019) pp. 101-114

Received: 18 Dec, 2018; Accepted: 19 March, 2019; Published: 25 April, 2019

Towards a Cooperative Intelligent System for Unpredictable and Predictable Road Hazard Detection

Islam Elleuch[1], Achraf Makni[2] and Rafik Bouaziz[1]

[1] University of Sfax, MIRACL Laboratory, Faculty of Economics and Management,
Airport Road, km 4, B.P. 1088, 3018, Sfax, Tunisia
islam.elleuch@fsegs.rnu.tn, rafik.bouaziz@usf.tn

[2] University of Sfax, MIRACL Laboratory, National School of Electronic and Telecommunication,
Tunis Road, km 10, B.P. 1163, 3018, Sfax, Tunisia
achraf.makni@enetcom.usf.tn

***Abstract*: Road hazards can cause dangerous accidents which lead to serious effects on human safety, destruction of vehicles and traffic flow disorder. Therefore, numerous systems based on *Vehicular Ad hoc Networks* (VANET) have been proposed to prevent this kind of accidents and enhance road users' safety. Nevertheless, these systems suffer from some problems, which can reduce their performance. For example, some of the proposed systems are autonomous; they do not exploit VANET to cooperate. In addition, as these systems offer more and more features, they treat a large amount of data, but without storage in a database. This can lead to the problems of congestion or loss of data. Therefore, we propose in this paper a new system, entitled *Cooperative Road Hazard Detection Persistent System* (CopRoadHazDPS). This system is based on the use of (i) *Vehicle-To-Vehicle* (V2V) and *Vehicle-To-Infrastructure* (V2I) communications to promote cooperation between vehicles, infrastructures and the Control Center, and (ii) *Real-Time DataBases* (RTDB) to manage data in real-time, effectively and accurately. It ensures the road safety in the case of unpredictable and predictable road hazards. Once a vehicle or the Control Center identifies a road hazard, its *Road Manager* cooperates with the other components to analyze the situation and decides about the convenient actions to avoid accidents. Simulations of several driving scenarios in varied type of roads, within the *Vehicles In Network Simulation* (VEINS) framework, confirm that CopRoadHazDPS ensures safety and reduces data freshness transactions thanks to the concept of the Quality of Data.**

Keywords: Cooperative road hazard detection, unpredictable hazard, predictable hazard, RTDB, VANET communications**.**

I. Introduction

Approximately, 1.3 million people die in road accidents every year (*i.e.*, about 3,287 deaths a day) according to the Association for Safe International Road Travel statistics [1]. More than 90% of these accidents are caused by human error due to lack of sleep, fatigue or inattentiveness of the driver. In addition, most of these accidents occur because of sudden or unexpected road hazards and the inability of the driver to control the vehicle.

To improve road safety, researchers and automobile manufacturers have appreciably focused on the development of *Advanced Driver Assistant Systems* (ADAS) [2][3]. In the context of intelligent vehicles and ADAS, road hazard detection remains a challenging task that must be performed in real-time, robustly and accurately. An ADAS assists the driver in its trajectory to detect road hazards (stopping or moving objects) using sensors such as radar, *Light Detection and Ranging* (LiDAR), laser or camera. It also warns the driver and/or takes control of the vehicle to prevent dangers. It uses a large number of sensors to obtain current information about the vehicles and their surrounding changes. However, ADAS based only on their sensors, called autonomous ADAS, are not efficient in some situations (*e.g.*, bad weather conditions); they are not able to detect obstacles easily. To enhance road safety, *Cooperative ADAS* (C-ADAS) are being proposed to manage roads using wireless communications [4]. The *Vehicular Ad Hoc Networks* (VANET) are the most appropriate networks for the *Intelligent Transport Systems* (ITS) [5]. They permit vehicles to transmit and receive information by *Vehicle-To-Vehicle* (V2V) and *Vehicle-To-Infrastructure* (V2I) communications [6]. Obviously, vehicles and the infrastructure must be equipped with the necessary devices; each vehicle uses a device installed at the dashboard, called *On Board Unit* (OBU), and the infrastructure uses devices, called *Road Side Units* (RSU). We note that a smart road must include many infrastructures, one by each portion. A vehicle communicates with the RSU of the nearest infrastructure, which communicates with the Control Center [7]. The latter has to receive information from the different RSU, analyze the collected information to determine the alerts to send for these RSU, which have to transmit the received alerts to vehicles in their range.

In the literature, we differentiate two categories of hazard identification systems in VANET: unpredictable and predictable hazard detection systems. Indeed, the unpredictable hazards occur at any road type and position, while the predictable hazards happen at known positions in the road (*e.g.*, intersection, highway). However, if two systems of these two categories operate independently and do not interact, they cannot guarantee a complete safety. In addition, the majority of the techniques used by these systems are limited to warn drivers in case of an obstacle or dangerous zone detection. Thus, the driver must react instantly, which is not always the case. Furthermore, these systems do not store any data in a database and consider only the current information; however, preceding information can improve the detection and analysis results.

To guarantee more secure and efficient road traffic management, we propose a new C-ADAS, called *Cooperative Road Hazard Detection Persistent System* (CopRoadHazDPS). To reach this goal, this system must ensure two activity types. First, it has to precisely reveal the current environment state. To do this, CopRoadHazDPS ought to (i) transmit, save, update and analyze continuously large amount of data obtained by means of V2V and V2I communications, and (ii) respect the time constraints related to the data validity and the transaction deadlines, which requires the use of *Real Time Data Bases* (RTDB) and an *RTDB Management System* (RTDBMS). Second, CopRoadHazDPS has to deal with both types of hazards: unpredictable and predictable. For this reason, it includes two sub-systems: unpredictable CopRoadHazDPS and predictable CopRoadHazDPS. These two systems coordinates between them and each one is triggered according to the situation. In our previous papers [8] to which we bring in this paper a major extension, we have proposed an unpredictable CopRoadHazDPS that integrates the functionalities of the unpredictable road hazard: obstacle detection and alert about dangerous zones. It relies on (i) VANET to establish communications between vehicles, infrastructures and the Control Center and (ii) *Real-Time DataBases* (RTDB) to manage data in real-time. We defined a distributed processing in the Road Manager of each vehicle to be able to detect and avoid sudden and unpredictable road hazards. The evaluations show the performance of unpredictable CopRoadHazDPS in assisting drivers and taking the suitable actions when identifying unexpected danger. In order to improve the safety of traffic management, we enhance in this paper CopRoadHazDPS by incorporating predictable road hazard functionalities. The predictable CopRoadHazDPS has to focuse on two applications: cooperative left turn assistance in T-intersection and cooperative on-ramp merging in highways. The RSU device, installed in each zone, transmits the collected data from vehicles in its range to the Control Center. This latter achieves a centralized processing, in its road manager, to predict and avoid collisions between vehicles in these dangerous zones.

The remainder of this paper is organized as follows. The next section reviews the related works on road hazard detection. In Section III, we describe the architecture of the proposed CopRoadHazDPS. Section IV describes the working principle of each sub-system of CopRoadHazDPS. Simulation results and a discussion are presented in Section V. Finally, Section VI concludes the paper and gives some future works.

II. Related works

Road hazard detection systems can be categorized into two classes according to the occurrence place of the hazard: (i) unpredictable road hazard detection if the hazard occurs unexpectedly at any place on the road and (ii) predictable road hazard detection if the hazard occurs at known places (*e.g.*, intersection, highway ramp, etc.). We examine in this section these two classes, unpredictable and predictable hazard detection. The aim is to understand how their systems function and to draw their issues and their inadequacies.

A. Unpredictable hazards detection

In [9], we have divided the unpredictable road hazard detection systems in two categories according to the type of road hazards: (i) Obstacle Detection (ObstDetect) if the hazard is a static (*e.g.*, a broken-down vehicle, a speed bump, debris, ice, etc.) or dynamic (*e.g.*,. unexpected dynamic vehicle, a pedestrian, an animal, etc.) object or (ii) Dangerous Zones Alert (DZonAlert) if the hazard is a construction zone, an accident or a traffic congestion. In fact, numerous contributions in the field of ObstDetect and DZonAlert have been proposed to guarantee the safe navigation of a vehicle. Most of the contributions generate audible and visual alerts to call the driver to adjust the vehicle's speed. In addition, the majority of the techniques of the ObstDetect and DZonAlert systems rely on single [9] or multiple sensors [10] and cameras for image or video processing [11]. However, relying on one or many sensors may remain ineffective in bad weather conditions (*e.g.*, ice, fog, snow and rain) due to sensors' limited detection capacity. Moreover, the parallel processing of frames (from camera) and scans (from LiDAR) is costly and time-consuming. As a result, the inter-vehicle communications have been introduced in the ObstDetect and DZonAlert systems.

Regarding the ObstDetect systems, in the GeoNet project, once a vehicle detects a black ice, the information is quickly forwarded as long as there are vehicles within the geographical area using V2V and V2I communications [12]. The proposed ObstDetect system in [13] combines windshield cameras, computer vision, V2V communications and laser holographic projection. The authors of [14] propose an application regarding to real-time image processing for VANET to identify obstacles.

As for the DZonAlert systems, the crashed vehicle sends an alert to the vehicles near the accident zone via V2V communications, and to the RSU of the near infrastructure via V2I communications [15]. This infrastructure transmits this alert to the other vehicles coming in the same path, via V2I communications, to ask them to change their direction for reaching their destination. The authors of [16] propose to analyze information gathered from neighboring vehicles through VANET communications, to predict accidents before they happen and alert vehicles in the case of an accident risk. Moreover, numerous researches in VANET focus on discovering and broadcasting traffic congestion information [17]. The proposed system in [18] detects the traffic congestion and disseminates warning messages from the affected vehicles to the nearby vehicles by V2V communications, and for long range by V2I communications. After receiving the warning, the follower vehicles try to avoid the traffic congestion.

HERE Hazard Warnings application [19], also based on rich vehicle sensor data, notifies drivers about potential road hazards in real-time. However, it only covers the flowing six cases: accident, broken-down, slippery road, fog and heavy rain warnings.

As for our contribution, we proposed in [8] a system that ensures the road safety in both ObstDetect and DZonAlert. Our proposed system covers any road hazard occurred in the road.

B. Predictable hazards detection

The predictable hazards detection applications assist the driver to discover dangers at fixed areas. These zones require more attention: they should be under control to ensure a safe and quite driving. We describe in the following of this sub-section contributions in two areas, regarding the predictable hazards: left turn assist and highway on-ramp merging.

- **Left turn assistance:** A left turn at a T-shape intersection is one of the most typical collisions involving a huge number of injuries and fatalities. *Left Turn Assistance* (LTA) system is a specific ADAS that supports the driver to make a safe left turn by means of a warning strategy [20]. The literature on this field of cooperative LTA systems in C-ADAS is not extensive. The authors in [21] investigated a collision detection algorithm for the T-shaped intersection based on the *Location Based Service* (LBS) technology. Each vehicle exchanges its derived information from LBS-based equipment to the adjacent vehicles and infrastructure via *Dedicated Short Range Communications* (DSRC). The system determines the possibility of collisions based on the comparison of the entering time and leaving time of the vehicle attempting to turn left and the oncoming vehicle to the conflict area. These times are calculated based on the geometric parameter of the intersection and the vehicle state information output by Kalman filter. The proposed study in [22] deals with unprotected left turn maneuvers at intersections. This study adopts a game theory based framework to capture the dynamic interactions between the conflicting vehicles in a connected environment. The vehicle attempting to effectuate the left-turn maneuver calculates the acceleration/deceleration rate required to avoid the collision. It considers a comfortable acceleration/deceleration rate to perform the left-turn maneuver safely.
- **Highway on-ramp merging:** The merging of on-ramp traffic flow with the incoming flow from the mainstream is a critical problem on highways. Numerous works focus on regulating the flow of vehicles merging into the highway in order to avoid collisions and decrease traffic congestion. In [23], the authors proposed an optimal trajectory planning methodology to assist the merging of vehicles. This solution aims to achieve safe and traffic-efficient merging while minimizing the engine effort and passenger discomfort. The authors in [24] presented an optimization framework and an analytical solution to coordinate vehicles and achieve a smooth traffic flow without stop-and-go driving at merging zones. The proposed solution allows avoiding collision while reducing fuel consumption. The proposed approach in [25] adopts a reinforcement learning algorithm to find an optimal merging policy.

C. Discussion

Treating hazard detection in ADAS with VANET is still in its primary phase. The above-mentioned systems operate separately and distinguish between unpredictable and predictable hazards. Indeed, we think that one system that combines these two sub-systems improves cooperation and increases performance better than two separate sub-systems. This integration would make possible the automation of the whole process of hazard detection and the treatment of the majority of unexpected situations. Thus, such a system is able to detect road hazards, send alerts and react autonomously to avoid a potential collision. It constitutes our first contribution in this paper.

Moreover, we notice that the studied systems are based on probabilistic or recursive calculations and do not manage data storage. In fact, they may expand the number of messages exchanges regularly between vehicles and cause data loss. In order to treat these issues, we propose here to integrate an RTDB in each vehicle and in the Control Center. To the best of our knowledge, the initiative of handling data based on RTDB storage has not been attempted for the same objective as in our work. It constitutes our second contribution in this paper.

III. The architecture and environment of CopRoadHazDPS

A. General description

The architecture we propose for our CopRoadHazDPS is shown in figure 1. The CopRoadHazDPS comprises the road infrastructure and the vehicle components. The different components of this system act as follows. The *Vehicle Monitor* transmits information about location, speed, and direction to the *Information Collector* (InfCollect) and to the OBU device mounted on the vehicle. InfCollect is responsible for feeding and updating the RTDB, through an RTDBMS, by new data. The OBU has to send the vehicle data to (i) the other OBUs of surrounding vehicles and (ii) the RSU of the nearest infrastructure through DSRC [26]. On the one hand, the received information by a vehicle from the neighboring vehicles is transferred to the InfCollect that saves these data in the local vehicle RTDB. On the other hand, the received vehicles information by the RSU is transmitted to the InfCollect of the Control Center that saves or updates these data in the RTBD of the Control Center. The *Hazard Detector* (HazDetec) of a vehicle is responsible for perceiving the road environment. It has to detect any road hazard using obstacle detection sensors or via a sudden change of the vehicle movement. When the HazDetec of a *Host Vehicle* (HostVeh) detects a road hazard, it adds this event, with its type, localization, lane and discovery instant, in the local RTDB. Simultaneously, the *Road Manager* (RoadManag) examines the situation and asks the *Overtaking Manager* (OvertManag) to check if it can overtake the road hazard or not. When the *Vehicle Motion Controller* (VehMotCont) of the HostVeh receives the commands from the RoadManag, it applies a braking to either decelerate or stop, or overtake the vehicle to avoid the collision. At the same time, the OBU of the HostVeh sends alerts about this road hazard to the vehicles driving on the same road behind it and in its range, called *Backward Vehicles* (BackVeh). As soon as it receives a warning, the RoadManag of each BackVeh analyzes the situation and its InfCollect inserts the road hazard information in its RTDB. The RoadManag achieves in parallel the two following actions: it (1) calculates the distance between the vehicle and the road hazard and/or the preceding vehicle; (2) asks the OvertManag to see if it is possible to overtake the road hazard or not. With the obtained results, the RoadManag decides about the appropriate actions to ensure safety. Then, it triggers the VehMotCont to perform these actions: change the vehicle lane and overtake, or brake to stop if it is not possible to overtake. In addition, the RoadManag queries periodically the RTDB to be informed about the safety of the road ahead and to react, if necessary, in the same manner. Moreover, vehicles transmit the road hazard information to the Control Center, via the different RSU, and receive from it alerts about far road hazards. So, congestion can be avoided as far vehicles can change their directions to achieve their destinations.

Adding to the unpredictable road hazards, CopRoadHazDPS can predict road collisions between vehicles at specific zones. In these zones, CopRoadHazDPS is strongly based on road infrastructure, composed by RSU devices and the Control Center.

Indeed, the position of each “predictable collision space” is stored in the Control Center’s RTDB. On the other hand, the RSU send the received vehicle data to the InfCollect of the Control Center, which saves or updates the acquired information in its RTDB. After that, the RoadManag of this center has to achieve an iterative process: it queries the RTDB to be informed about the current situation. If there are at least two vehicles from two different directions near the “predictable collision space”, it triggers the *Situation Manager* (SitManag) to analyze this situation, *i.e.*, to determine whether these vehicles will occupy the same “predictable collision space” at the same time. If so, the RoadManag sends alerts to the concerned vehicles. Therefore, the VehMotCont of the lower priority vehicle (*i.e.*, coming from the secondary road) reacts to stop or decelerate it. The higher priority vehicle (*i.e.*, coming from the primary road) continues its trajectory without decelerating or stopping. It’s the RoadManag of the Control Center that is responsible for assigning the vehicles priority by consulting the road cartography. When the RSU detects that the higher priority vehicle skips the “predictable collision space”, it informs the InfCollect. Then, the RoadManag retriggers again the SitManag to analyze the new situation.

B. Data model

We symbolize each vehicle by a node specified by its position (X, Y), its speed ($V_{x,y}$) and its flow direction (Dir) [27]. We suppose that a *Global Positioning System* (GPS) device is installed on each vehicle. Note that the coordinates X and Y of vehicles and road hazard are obtained approximately from the GPS device with accuracy between 0.5 m and 1m.

On the one hand, an RTDBMS manages these data with their time stamps and validity times in order to be able to distinguish valid data from obsolete ones. So, we represent the data structure as a triplet d = (d_{value}, $d_{timestamp}$, $d_{duration}$), according to our data diagram presented in [27]. d_{value} denotes the recorded value of the data, $d_{timestamp}$ indicates the time of the last obtained data and $d_{duration}$ designates the absolute validity time of the data item.

On the other hand, each vehicle disseminates its speed, position and direction in separate periodic messages because these pieces of information do not have equal validity times. The data delivery process is based on the validity time of data. Therefore, each piece of information has to be transferred to the InfCollect just before the end of the validity time of its previous version in the RTDB, in order to improve the data freshness. Each speed, position or direction message begins with the identification number (V_ID) of the corresponding vehicle and ends with the time (Time) when the data is obtained. In addition, a speed message contains the vehicle’s speed, a position message contains the position coordinates taken from the GPS, and a direction message contains the vehicle’s direction. As for an unpredictable hazard message, it contains its type (*e.g.*, obstacle, accident etc.), its position, its lane and the instant of its detection. Moreover, the position and the lane of each predictable hazard zone are stored in the RTDB of the Control Center, for predictable CopRoadHazDPS processing.

Thanks to the concepts of *Quality of Data* (QoD), the InfCollect of a vehicle does not have to systematically update its RTDB as there is a *maximum error tolerated* between the current value of each data item and its previous value stored in the RTDB, called the *Inaccuracy Threshold* (InacThr). Therefore, the update of a data value should be executed only if the difference between the obtained value and the value stored in the RTDB is higher than its InacThr. Consequently, both the number of ‘write’ operations and the conflict risk between transactions on the RTDB are reduced. Thus, transactions can further respect their deadlines. We determine the values of the InacThr by simulations according to driving conditions. Similarly, sending messages only occurs when the update of the concerned data value is necessary. In this case, two actions are executed in parallel: the InfCollect of the vehicle updates the data value, and the OBU transmits the message containing this data value to the neighborhood vehicles and to the RSU. Otherwise, the OBU does not forward the corresponding data message. Consequently, the number of transmitted messages is also reduced.

IV. CoopRoadHazDPS operating principle

The unawareness of a sudden road hazard during driving is the main reason for a collision. An accident can be avoided when the vehicle knows about the road hazard as soon as possible to react at time. In addition, the vehicle has to inform its neighbor vehicles about the road danger to avoid a chain of collisions.

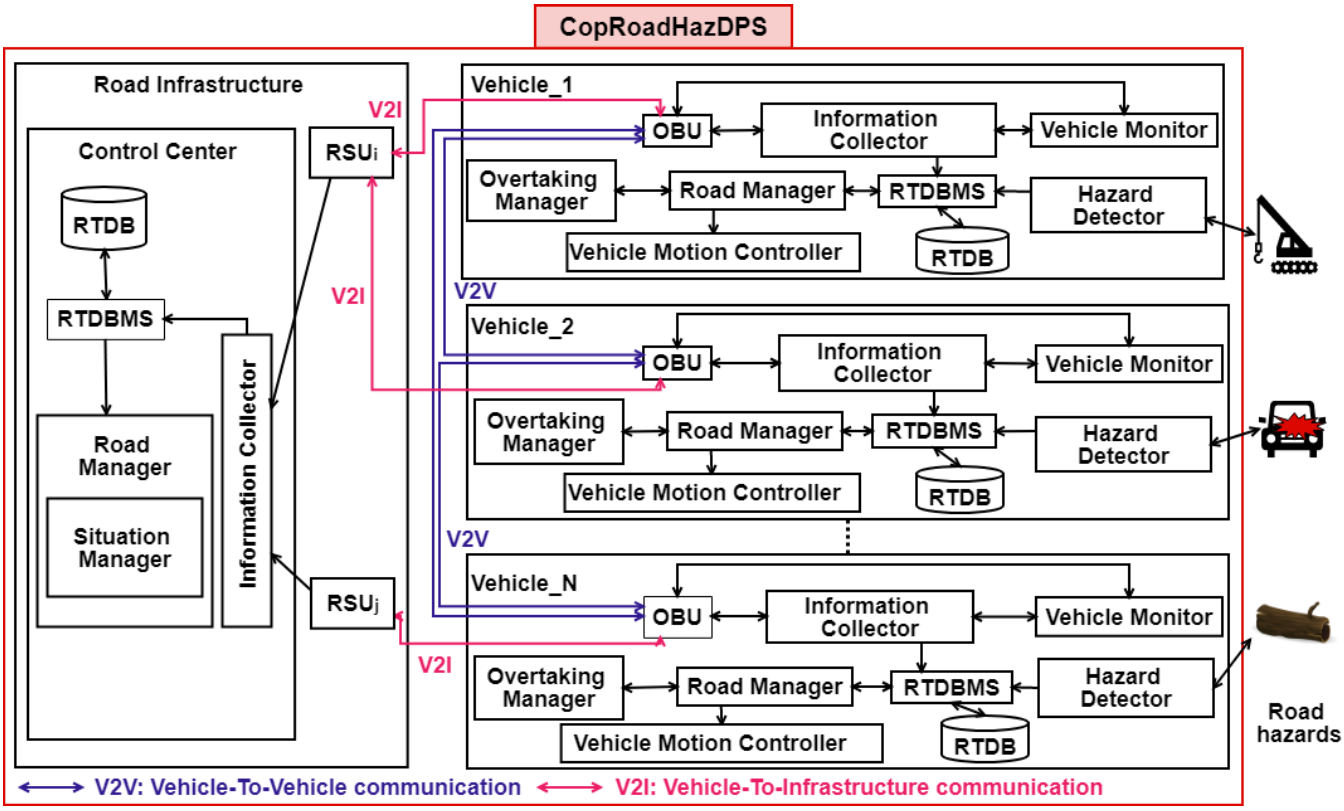

Figure 1. Architecture of CopRoadHazDPS

CopRoadHazDPS offers great capabilities to inhibit critical accidents and preserve human lives. It is based on V2V and V2I communications where every vehicle creates a connection with other vehicles and the RSU in its range.

As we assume an automated driving mode, in which each vehicle is equipped with automatic actuators, CopRoadHazDPS gives the commands to move or stop, accelerate or decelerate, and the vehicle reacts without the driver's supervision.

CopRoadHazDPS has the ability to detect road hazards in unpredicted and predicted places. In the following, we describe the CopRoadHazDPS functionalities in its two sub-systems: Unpredictable CopRoadHazDPS and Predictable CopRoadHazDPS.

A. Unpredictable CopRoadHazDPS

The treatment of the unpredictable CopRoadHazDPS is done in the vehicle as it is faster than receiving orders from the infrastructure. The HazDetec of a HostVeh is responsible for revealing the *Unpredictable Hazard* (UnpredHaz) via the vehicle's sensors, the nearest infrastructure or a sudden movement change of a preceding vehicle, for example.

Figure 2 shows a case of an unpredictable hazard. Once a HostVeh discovers an UnpredHaz, the HazDetec adds its information in the HostVeh's RTDB.

At the same time, the HostVeh's RoadManag examines the situation to take the suitable decision. It triggers the OvertManag to check whether the HostVeh can switch the lane to overtake the UnpredHaz and have enough time to return back to its original lane. Therefore, the VehMotCont reacts automatically according to the decision received from the RoadManag. It achieves one of these two actions. In case of no risk to overtake, the VehMotCont changes the HostVeh lane to avoid a collision. In case of an overtaking risk, the VehMotCont applies a braking to stop the HostVeh.

In addition, as soon as the HazDetec of the HostVeh discovers an UnpredHaz, the OBU notifies the Control Center via the nearest RSU, using V2I communications, and the adjacent BackVeh, using V2V communications. The InfCollect of the Control Center and those of these BackVeh insert in their RTDB the UnpredHaz information.

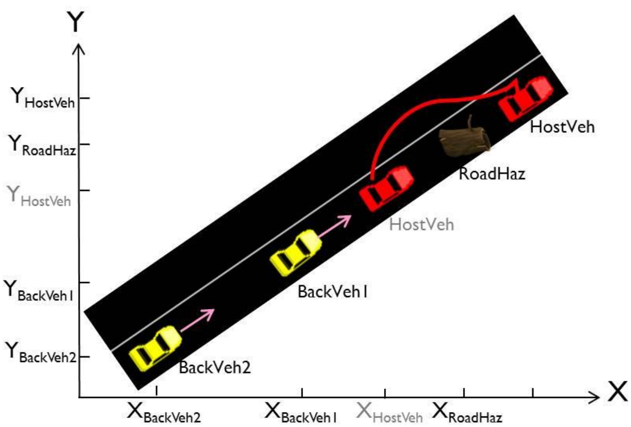

Figure 2. Example of a scenario in which an unpredictable hazard occurs on the road

The first BackVeh's RoadManag analyzes the situation in the same manner of those of the HostVeh's RoadManag, and transmits the appropriate commands to the VehMotCont. If the HostVeh stops, the first BackVeh must also stop before arriving to the HostVeh in order to ensure that no collision will happen between these two vehicles. Therefore, the BackVeh has to maintain the safe distance with the HostVeh. This distance is maintained if the distance between HostVeh and

BackVeh (BackHostDist) is superior to the safe distance of the BackVeh (SD_{Back}), as displayed in the following condition:

$$BackHostDist >= SD_{Back} \quad \text{(C1)}$$

The RoadManag calculates the safe distance of its vehicle *Veh* using the following formula F1 [28]:

$$SD_{Veh} = \left(CV_{Veh} / 10\right) \times 3 \quad \text{(F1)}$$

where CV_{Veh} is the current speed of the vehicle.

The distance between two vehicles, here between BackVeh and HostVeh (BackHostDist), is calculated thanks to the Euclidian distance formula:

$$\text{BackHostDist} = \sqrt{(X_{Host} - X_{Back})^2 + (Y_{Host} - Y_{Back})^2} \quad \text{(F2)}$$

where X_{Host} and Y_{Host} correspond to the coordinates (X, Y) of the HostVeh, and X_{Back} and Y_{Back} are the coordinates (X, Y) of the BackVeh.

If the distance BackHostDist is less than the safe distance of the BackVeh, the Road-Manag of this later triggers its VehMotCont to decelerate or to stop the vehicle, as represented by the rule R1

$$\textit{if } (\text{BackHostDist} < SD_{Back}) \quad \text{(R1)}$$
$$\textit{then } \text{VehMotCont is triggered}$$

Finally, the first BackVeh transmits, in its turn, a message containing its decision (overtaking or stopping) to the other BackVeh, which act in the same manner to avoid a chain collision.

We note that the RoadManag of each vehicle has to calculate a derived data, *i.e.*, the acceleration/deceleration. We define the formula F3 to calculate this data value.

$$a = \frac{CV_{Veh} - SV_{Veh}}{Ct - Pt} \quad \text{(F3)}$$

$$\text{with } Pt = \max\left(St,\ \left(Ct - PSV_{Veh}\right)\right)$$

where CV_{Veh} correspond to the current value of the vehicle *Veh* speed, while SV_{Veh} is the previous speed value stored in the RTDB of the vehicle *Veh*; Ct, Pt and St refers to the current time of CV_{Veh}, the previous time and the time of SV_{Veh} in the RTDB, respectively; PSV_{Veh} is the length of the validity time interval of SV_{Veh}; (Ct - PSV_{Veh}) is the virtual time of SV_{Veh}, used when it was not necessary to update a previous value of the vehicle speed thanks to the concepts of QoD.

The vehicle *Veh* is speeding down if *a* is negative, *i.e.*, (CV_{Veh} - SV_{Veh}) is negative. Upon receiving a new speed message from its preceding vehicle, the RoadManag of the concerned vehicle (*i.e.*, whatever the first BackVeh or the other BckVeh) compares the received speed value and the last stored value in its local RTDB. So, it verifies if the preceding vehicle decelerates or not through the condition C2.

$$CV_{Veh} < SV_{Veh} \quad \text{(C2)}$$

Note that the calculation of acceleration/deceleration in other systems is complex because they do not have any stored data, and so, they cannot apply the Formula F3.

B. *Unpredictable CopRoadHazDPS*

We focus in this sub-section on the description of the two new sub-systems in the context of predictable CopRoadHazDPS related to known and fixed places: cooperative turn left assistance at T-intersections and on-ramp merging at highways. The processing of these two sub-systems is centralized and achieved in the Control Center. To do so, an RSU device is installed next to each place: T-intersection (*cf.* figure 3 (a)) and on-ramp in highway (*cf.* figure 3 (b)).

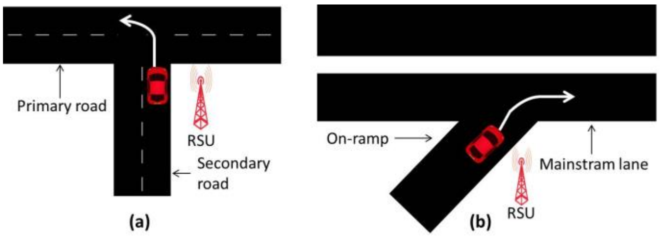

Figure 3. A model of a T-intersection (a) and a model of an on-ramp in highway (b)

1) *Cooperative Left Turn Assistance*

We describe in this sub-section the new cooperative left-turn assistance sub-system in the T-intersection. A T-intersection (or T-Junction) is a type of road intersection where a primary road meets a secondary road at right angles (or close to a right angle). So, the two roads form a T shape. The left-turn in a T-intersection is one of the most dangerous maneuvers.

We consider the case where a *Host Vehicle* (HostVeh) is driving in a secondary road and will perform a left-turn maneuver, while at last a *Target Vehicle* (TargVeh) is driving in a primary road. The RoadManag of the Control Center maintains two lists, L_Dir1 and L_Dir2, one for each provenance direction of the target vehicles towards the T-intersection. An impending collision may occur when the HostVeh and a TargVeh will occupy, at the same time, the same space. We call this "predictable collision space" a *Virtual Collision Point* (VCP) (*cf.*, figure 4), calculated by the RoadManag of the Control Center. So, the InfCollec of this latter monitors the vehicles in the range of the nearest RSU to the T-intersection, and keeps track of data in the Control Center's RTDB. When this RoadManag detects a vehicle near the T-intersection coming from the secondary road, it determines the HostVeh movement intention. As soon as it detects that the HostVeh attempts to turn left, the RoadManag launches an iterative process and checks if there are at least nonempty list, L_Dir1 or L_Dir2, as represented by the rule R2.

$$\textit{if } (L\_Dir1 \neq \phi \textit{ or } L\_Dir2 \neq \phi) \quad \text{(R2)}$$
$$\textit{then } \text{RoadManag triggers SitManag}$$

If the condition of R2 is false, *i.e.*, the two lists are empty, the RSU transmits a message to the HostVeh to inform it that it can turn left safely. Otherwise, the SitManag of the Control Center analyzes the situation and predicts any possibility of a collision. So, it determines the head(s) vehicle(s) of L_Dir1 and/or L_Dir2 (*i.e.*, the coming vehicles closest to the T-intersection from the two directions of the primary road).

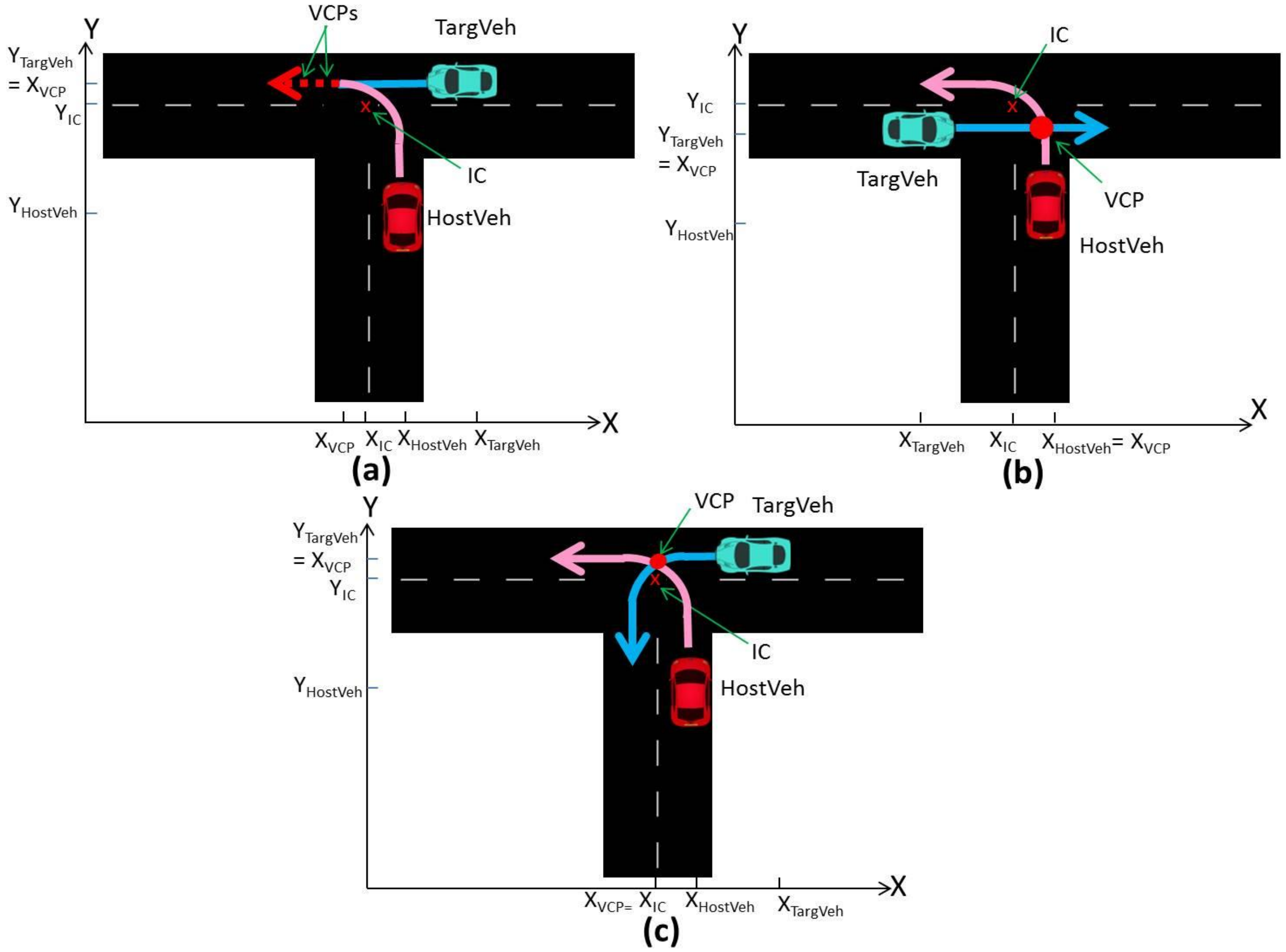

Figure 4. Example of a scenario in which a predictable hazard occurs at the T-intersection while turning left

Then, it calculates the two distances of the HostVeh and TargVeh to the first VCP when there are several, referred to as D1 and D2, respectively. These two distances are calculated by the Euclidian distance formula, as shown by formulas F4 and F5, respectively.

$$D1 = \sqrt{(X_{HostVeh} - X_{VCP})^2 + (Y_{HostVeh} - Y_{VCP})^2} \qquad \text{(F4)}$$

$$D2 = \sqrt{(X_{TargVeh} - X_{VCP})^2 + (Y_{TargVeh} - Y_{VCP})^2} \qquad \text{(F5)}$$

where $X_{HostVeh}$ and $X_{TargVeh}$ are the coordinates X, and $Y_{HostVeh}$ and $Y_{TargVeh}$ are the coordinates Y of the HostVeh and TargVeh. X_{VIP} and Y_{VIP} refer to the coordinates (X, Y) of the VCP.

Then, the SitManag calculates the required duration time from both HostVeh and TargVeh to reach this point, $T_{HostVeh}$ and $T_{TargVeh}$, based on formulas F6 and F7.

$$T_{HostVeh} = \frac{D1}{CV_{HostVeh}} \qquad \text{(F6)}$$

$$T_{TargVeh} = \frac{D2}{CV_{TargVeh}} \qquad \text{(F7)}$$

where D1 and D2 correspond to the distances between the HostVeh and VCP and between TargVeh and VCP, respectively. $CV_{HostVeh}$ and $CV_{TargVeh}$ are the current speed of the HostVeh and TargVeh, respectively.

It is crucial that both HostVeh and TargVeh have enough distances to continue their path without any damage. To do so, the SitManag compares $T_{HostVeh}$ minus $T_{TargVeh}$ to a certain *Threshold* (T_{th}). Only when ($T_{TargVeh}$ - $T_{HostVeh}$) is higher or equal to T_{th}, the HostVeh has sufficient time to turn left securely, as represented by rule R3.

$$\textit{if} \ (T_{TargVeh} - T_{HostVeh} >= T_{th}) \qquad \text{(R3)}$$
then VehMotCont of HostVeh is triggered to pass

Otherwise, there is a risk of collision between the HostVeh and TargVeh. So, the HostVeh must decelerate or stop, as displayed by rule R4, while the TargVeh crosses the T-intersection and continues its trajectory without stopping as it is coming from the primary road.

$$\textit{if} \ (T_{TargVeh} - T_{HostVeh} < T_{th}) \qquad \text{(R4)}$$
then VehMotCont of HostVeh is triggered to decelerate or stop

In fact, it is the RSU that sends the SitManag decision to the OBU of the HostVeh; so, the MotCont of this latter commands the HostVeh to pass or, conversely, to decelerate or to stop in order to avoid the crash. Moreover, the *Backward Vehicles*

(BackVeh) of the HostVeh also must decelerate or stop to maintain the safe distance and avoid a chain of collision (*cf.* rule R1) as explained in the unpredictable CopRoadHazDPS.

After the TargVeh skips the VCP, the RoadManag of the Control Center triggers again its SitManag to (i) verify whether the primary road near the T-intersection is occupied by other vehicles or not, and (ii) continue its process in the same manner.

Let us now take the illustrative examples of figure 4 to show how the SitManag calculates the coordinates of (X_{VCP}, Y_{VCP}) of the VCP. It determines these coordinates according to the movement intention of the TargVeh and HostVeh, on the one hand, and to the *T-Intersection Center* (IC) and the road *Widths* (W_p for the primary road and W_s for the secondary road), on the other hand. The Control Center's RoadManag gets the coordinates information of the IC from the road map. Figure 4 gives three scenarios where the HostVeh always turns left, from the South to the West.

In the first scenario, illustrated by figure 4 (b), the TargVeh is driving from the West to the East. The coordinates of VCP are calculated in this manner:

$$X_{VCP} = X_{IC} + \frac{1}{4} W_s \quad \text{(F8)}$$

$$Y_{VCP} = Y_{IC} - \frac{1}{4} W_p \quad \text{(F9)}$$

In the second scenario, illustrated by figure 4 (c), the TargVeh is driving from the East to the South. The coordinates of VCP become as follows:

$$X_{VCP} = X_{IC} \quad \text{(F8')}$$

$$Y_{VCP} = Y_{IC} + \frac{1}{4} W_p \quad \text{(F9')}$$

As for the third scenario, illustrated by figure 4 (a), the TargVeh is travelling from the East to the West. If in the two first scenarios, HostVeh and TargVeh share only one VCP, they share in this third scenario more than one VCP along their trajectory, as they will take a same new direction. However, it is sufficient that the SitManag operates as previously using only the first VCP. The coordinates of the first VCP are calculated using the formulas F8' and F9', respectively.

As for the case where the HostVeh will perform a right-turn maneuver, it is simple to deal with. Indeed, only one scenario is problematic, it is the one where the TargVeh is traveling from the west to the east when taking the topology of figure 4. The cooperative assistance sub-system in the T-intersection operates in a same manner as for the left-turn, but only controls the west-east direction and uses only F8 and F9 to calculate the VCP coordinates.

Finally, we discuss two other cases:

- Case where there is a stop sign at the end of the secondary road of the T-intersection. The HostVeh must mandatory Stops. The cooperative assistance sub-system in this type of T-intersection operates in a simpler manner; it is limited to giving a start permission to the HostVeh when there are no target vehicles near the VCP in the primary road.
- Case of an imperfect T-intersection, where the direction of the secondary roads is not parallel to the Y-axis. If cooperative assistance sub-system in this type of T-intersection operates in a similar manner, but the formula F8 must be revised using trigonometry adjustments to be suitable for the T-intersection topology. We just give here the following adjustment example relating to the case of figure 5. Suppose β measures the inclination angle between the x-axis and the direction of the secondary road. F8 becomes as follows; β takes here a certain value ($0 < \beta < \pi/2$).

$$X_{VCP} = X_{IC} + \sin\beta \times \frac{1}{2} W_s \quad \text{(F8)}$$

2) Cooperative highway on-ramp merging

The working principle of the cooperative merging on a highway is similar to the cooperative right turn assistance. We have here a single mainstream highway lane, with a single-lane on-ramp, as illustrated in figure 6.

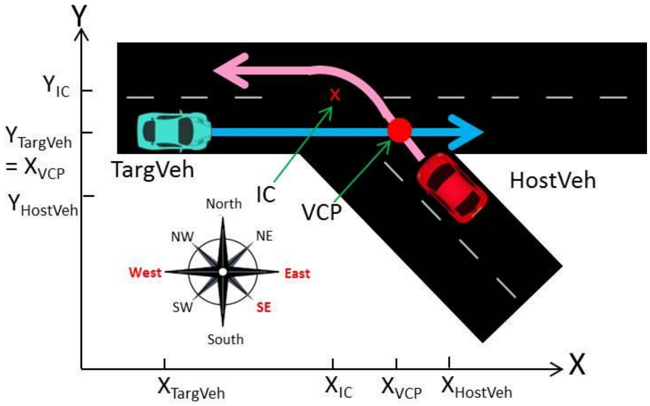

Figure 5. Example of a turn left scenario in an imperfect T-intersection

We deal with the case of a merging vehicle, entitled *Host Vehicle* (HostVeh), which comes from the on-ramp. This HostVeh will perform the merging maneuver to enter the mainstream flow, while a *Target Vehicle* (TargVeh) is traveling in the mainstream lane. The RoadManag of the Control Center maintains here only one list, entitled L_Dir, that contains the vehicles coming from the mainstream lane. An imminent collision can occur when a HostVeh and a TargVeh will occupy, at the same instant, the same merging space. We define this fix predictable merging space as a *Virtual Collision Point* (VCP), as illustrated in figure 6. The VCP is close to the end of the on-ramp. Its coordinates (X_{VCP}, Y_{VCP}) are determined from the road card and stored in the Control Center's RTDB by this RoadManag.

When the Control Center's RoadManag identifies a vehicle coming from the on-ramp too close to the merging space, it triggers an iterative process which begins by checking the state of the list L_Dir using the rule R5.

$$\textit{if} \quad L\_Dir \neq \phi \quad \text{(R5)}$$

then RoadManag triggers SitManag

else Authorization to merge

If the condition of R5 is false, *i.e.*, the list L_Dir is empty, the RSU informs the HostVeh that it can merge onto the mainstream lane securely. Otherwise, the SitManag analyzes the situation and checks whether there is a risk of a collision. It determines the head vehicle of L_Dir (*i.e.*, the closest coming vehicles from the mainstream lane). Then, it calculates (i) the distance D1 between the HostVeh and the VCP (*cf.* formula F4), and (ii) the distance D2 between the TargVeh and the

VCP (*cf.* formula F5). After that, the SitManag calculates the required duration time from both HostVeh and TargVeh to reach this point, *i.e.*, $T_{HostVeh}$ (*cf.* formula F6) and $T_{TargVeh}$ (*cf.* formula F7). To continue their trajectory without any risk of collision, the HostVeh and TargVeh must either have sufficient times to do it, similarly to the cooperative right turn assistance sub-system, or will do not take the same lane or contiguous lane. To verify the time condition, the SitManag compares $T_{HostVeh}$ minus $T_{TargVeh}$ to a certain *Threshold* (T_{th}). If the rule R3 is satisfied, the VehMotCont of the HostVeh performs the required maneuvers to merge. Otherwise, *i.e.*, it is the rule R4 that is satisfied, the VehMotCont decelerates or stops the HostVeh to avoid the accident with the TargVeh. Moreover, the HostVeh informs its BackVeh about its deceleration to decelerate or stop and maintain the safe distance (*cf.* rule R1).

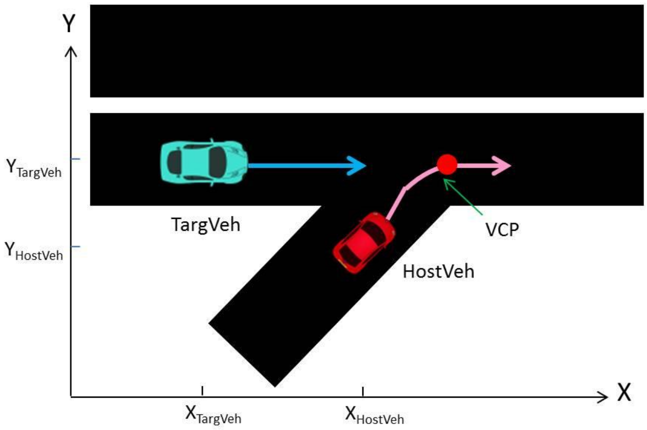

Figure 6. Example of a scenario in which a predictable hazard occurs at the on-ramp highway merging

V. Simulation and results

In this section, we proceed to evaluate, by simulation tasks, the performance of CopRoadHazDPS that combines (i) the functionalities of unpredictable and predictable road hazard detection and (ii) RTDBMS. Indeed, we are not currently able to deploy and test CopRoadHazDPS in a real-world environment, with real cars, requiring high cost and manpower. However, we try to comply with such an environment.

A. Simulation environment

We validate the C++ methods we have implemented for the formulae, conditions and rules defined in section 4 as well as the different components of CopRoadHazDPS, under *VEhicles In Network Simulation* (VEINS) framework [29]. VEINS is an open source vehicular network framework and is very close to reality. It bi-directionally couples the road traffic simulator Simulation of Urban MObility (SUMO) [30] and the discrete event-based network simulator OMNET++ [31], through TCP connections. It is able to generate a real-time interaction between road traffic and network simulators and supports the simulation of wireless communication protocols in VANET. We have also used SQLite3 to implement the different RTDB in the vehicles and the Control Center of CopRoadHazDPS. Then, we have tested different scenarios by changing the number of vehicles between two and ten vehicles. We set the maximum speed of each vehicle to 30 m/s in urban roads and to 40 m/s in highways. We deploy RSUs in each infrastructure portion and next to the T-intersections and the on-ramp in highways. We consider that the transmission range is at proximity of 500 meters for all vehicles and from the RSU. To establish V2I and V2V communications, we adopt the IEEE 802.11p *Wireless Access in Vehicular Environments* (WAVE) standard in DSRC [26].

B. Simulation results

In this section, we present the results of the simulations that validate the different components of CopRoadHazDPS.

Let us begin by showing how CopRoadHazDPS respects the validity time interval of each data item and the QoD principle. We start by fixing the length of the validity time interval of each data item in this way: (i) 1 s for the vehicle speed, (ii) 100 ms for its position (X, Y) and (iii) 5 s for its direction. Then, we define the value of *Inaccuracy Threshold* (InacThr) for speed (IT-S) to 1m/s and the InacThr for position (IT-P) to 1 m in order to reduce the number of update transactions. To better explain when the InfCollect executes the update transactions, we consider an example where two vehicles V_1 and V_2 are driving on the road. The acquired speed data values for both vehicles V_1 and V_2 at time t = 52 s are 23.5 m/s and 14 m/s, respectively. Table 1 displays two cases of updating data using an RTDB and the QoD concept. For V_1, the difference between “the collected speed value (= 23 m/s) at time t = 53 s” and “the stored value in the RTDB” is less than the value of IT-S. Thus, this value is not updated and the system uses the value stored in the RTDB at time t = 52 s. But for V_2, the difference between “the acquired speed value (= 16 m/s) at time t = 53 s” and “the stored value in the RTDB” is higher than the IT-S value. So, the stored value in the RTDB at time t = 52 s is changed by the value obtained at t = 53 s.

Table 1. Updating data in CopRoadHazDPS using an RTDB at t =53 s

Data	Values	Timestamp	Duration
V_1_Speed	23.5 m/s	52	one second
V_2_Speed	~~14 m/s~~ 16 m/s	~~52~~ 53	one second

We evaluate now the efficiency of the two sub-systems of CopRoadHazDPS. When the RoadManag of a specific vehicle or of the Control Center identifies a risk with a road hazard, the VehMotCont must react as soon as possible to avoid a collision. To do so, we define a crucial parameter, that we call the reaction time (T_{react}). T_{react} is the time between instant when the HazDetect of a vehicle or the RoadManag of the Control Center detects a road hazard until the instant when the VehMotCont of this vehicle receives the order and reacts.

To estimate correctly T_{react}, we perform diverse simulations and calculate the average value of T_{react}. So, we realize several scenarios in which vehicles are faced to unpredictable or predictable road hazards. In these scenarios, we let CopRoadHazDPS alternatively:

- send messages (i) taking into account a data validity time for each data item, or (ii) taking into account also the QoD by varying the values of the InacThr for position and speed,
- consider different values of speed (maximum 30 m/s in urban roads and maximum 40 m/s in highways) and varied number of vehicles (from 2 to 10).

For the conducted simulations, we obtained an average value of T_{react} equal to 2.2 ms. As the typical reaction time for a human driver is estimated to one second [32], we see that COAPS enhances the reaction time.

Table 2. The values of different parameters to estimate T_{react}

Scenario	Speed (m/s)	Objective	Number of vehicles	InacThr	T_{react} (ms)
Sc1	40	On-ramp merging	2	Without	2
Sc2	20	Static obstacle detection	2	With	1
Sc3	10	Left turn assistance	3	Without	3
Sc4	30	On-ramp merging	3	With	2
Sc5	15	Traffic congestion alert	4	With	1
Sc6	15	Left turn assistance	5	Without	4
Sc7	20	Alert about accident	6	Without	3
Sc8	35	On-ramp merging	7	With	1
Sc9	20	Left turn assistance	9	Without	3
Sc10	25	Dynamic obstacle detection	10	With	2

In order to show the benefits of using RTDB, we define the following metrics:

- **The number of update transactions:** For the case of the distributed processing (*i.e.,* unpredictable CopRoadHazDPS), this number is the sum of the number of the local update transactions (equal to the number of sent messages by one vehicle) in a given vehicle's RTDB and the number of the external update transactions in the nearby vehicles' RTDB. For the case of the centralized processing (*i.e.,* predictable CopRoadHazDPS), it represents the number of update transactions in the RTDB of the Control Center.
- **The number of exchanged messages:** It is the total number of messages between vehicles and/or infrastructure via V2V and/or V2I communications.
- **The message loss ratio:** It is the ratio between the number of the lost messages and the total number of exchanged messages in the network.

We will display in the next sub-sections the evaluation results for two main scenarios: unpredictable and predictable hazard detection. For each scenario, we vary the number of running vehicles from two to ten, and we realize two experiments:

- we let CopRoadHazDPS send messages with taking into account the validity time for each data item (100 ms for position, 1 s for speed and 5 s for direction),
- we let CopRoadHazDPS send messages also taking into consideration the notion of QoD; using different values of InacThr for both position and speed data.

1) Unpredicted hazard scenario

Let us present in this sub-section the simulation results of the Unpredictable CopRoadHazDPS considering many types of unpredictable road hazards. We begin by verifying if the RoadManag of the vehicle identifies the unexpected road hazards successfully and prevent the collision. Then, we calculate the above-mentioned metrics without InacThr and with IT-S and IT-P.

To start with, our experimentations confirm that the Unpredictable CopRoadHazDPS operates accurately and avoids potential collisions with road hazards with the use of an RTDB in each vehicle.

For the next of this sub-section, we present the simulation results of a specific scenario where vehicles occurs an unexpected static obstacle and some of which can overtake safely the road hazard while others do not.

Figures 7 and 8 show the advantage of using the data validity for data freshness on the reduction of (i) the numbers of total update transactions in all the RTDB of the running vehicles (*cf.* figure 7), and (ii) the numbers of exchanged messages between vehicles and between vehicles and their nearest RSU (*cf.* figure 8), respectively.

In the first experiment, we let the Unpredictable CopRoadHazDPS use only the data validity of each information type. The simulation results are displayed by the red charts which confirm that the numbers of the total update transactions (*i.e.,* local and external updates) and numbers of exchanged messages and are immense. For the case of ten running vehicles for example, the number of update transactions is about 12840 (*cf.* figure 7, red chart) and the number of exchanged messages is more than 2220 (*cf.* figure 8, red chart).

In the second experiment, we let the Unpredictable CopRoadHazDPS use different values of InacThr for position and speed data. Figure 7 illustrates that the numbers of update transactions and the numbers of exchanged messages are remarkably reduced by introducing the concept of QoD. With an IT-P equal to 3 m and an ITS equal to 2 m/s, the number of update transactions is reduced to about 2900 transactions (*cf.* figure 7, orange chart) and the number of exchanged messages is reduced to about 980 messages (*cf.* figure 8, orange chart), for the example of ten running vehicles.

In fact, data have not been updated periodically in the RTDB of each vehicle. Contrariwise, data have been updated sporadically according to the different values of InacThr.

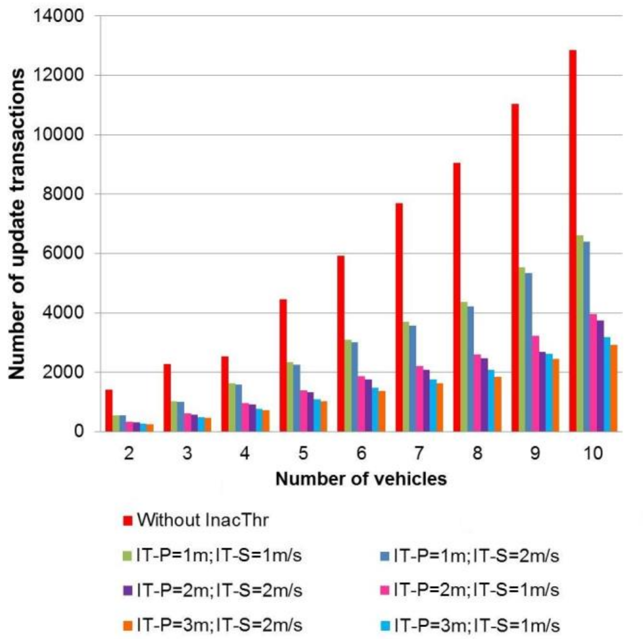

Figure7. Comparison of the numbers of total update transactions without InacThr, versus with IT-S and IT-P in the unpredicted hazard scenario

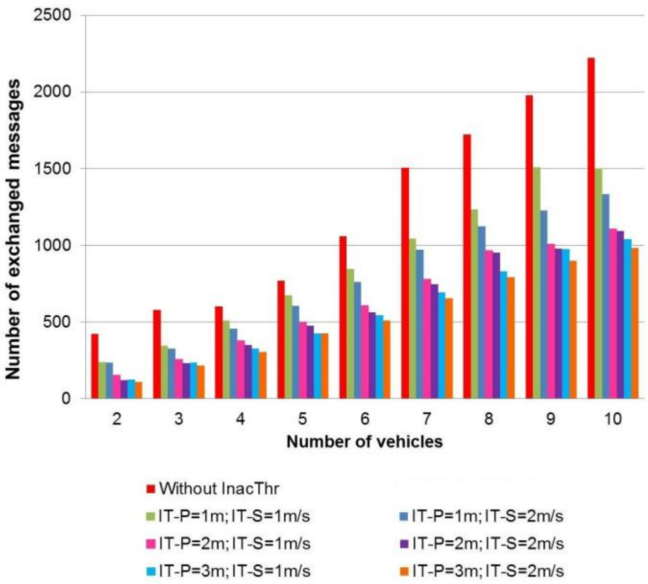

Figure 8. Comparison of the numbers exchanged messages without InacThr, versus with IT-S and IT-P in the unpredicted hazard scenario

As mentioned in the sub-section 3.2, the OBU of a vehicle transmits a message only when the InfCollect updates the data value in the RTDB. Therefore, the number of sent messages of a specific vehicle is decreased, and thus the number of exchanged messages is reduced.

Figure 9 displays the impact of the use of the QoD in the distributed RTDB on the message loss ratio, for an example where IT-S is equal to 2 m/s and varied IT-P values. Taking the case of ten running vehicles, the message loss ratio is 77 % (red chart) without the use of the InacThr for position and speed data. This ratio is decreased with the use of different values of the InacThr (the other charts): it is 56 % when IT-S is equal to 2 m/s and IT-P is equal to 3 m, for the same case. We notice a similar enhancement with I-TS equal to 1 m/s. Consequently, the concept of QoD reduces the risk of lost messages, and therefore improves the performance of the Unpredictable CopRoadHazDPS under VANET. Even with lost messages, the simulations confirm the effectiveness of the Unpredictable CopRoadHazDPS for avoiding accidents. Even if this amelioration resembles not enough, we remark that there are other general problems, not specific to the road hazard risks, to deal with in order to further decrease the amount of lost messages in VANET [33, 34].

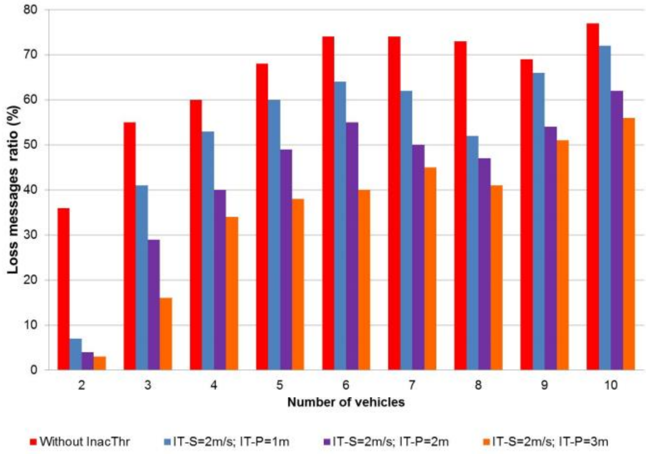

Figure 9. Variation of the ratio of loss messages, without InacThr versus with IT-S=2 m/s and varied IT-P in the unpredicted hazard scenario

2) Predicted hazard scenario

We now present the simulation results of the Predictable CopRoadHazDPS considering the two sub-systems: cooperative left turn assistance and cooperative highway on-ramp merging. Here also we start by checking if the RoadManag of the Control Center prevents the collision between two or more vehicles. Then, we calculate the metrics about the numbers of update transactions, and the numbers of exchanged messages, without and with the concept of QoD.

Our experimentations confirm that the Control Center, via its different components and its RTDB, assists successfully the vehicles during their maneuvers about left turn (and we deduce the same thing about right turn, which is simpler) in a T-intersection and merging in the on-ramp highway.

The simulation results of the cooperative left turn assistance scenario, as illustrated in figures 10 and 11, display that (i) the numbers of the update transactions in the RTDB of the Control Center, and (ii) the numbers of the exchanged messages between vehicles and between vehicles and the RSU of their range, are significantly reduced by the use of the concept of QoD.

In the case of ten vehicles, the number of update transactions is reduced from more than 1880 (*cf.* figure 10, red chart) when sending messages taking into account only the data validity of each data item (first experiment) to about 370 (*cf.* figure 10, orange chart) with an IT-P equal to 3 m and an IT-S equal to 1 m/s (second experiment), for example. In the same way, the number of exchanged messages is decreased from more than 2800 (*cf.* figure 11, red chart) to about 1260 (*cf.* figure 11, orange chart) with an IT-P equal to 3 m and an IT-S equal to 1 m/s, for example.

In fact, since data have been updated sporadically in the RTDB of the Control Center using different values of InacThr, the number of exchanged messages between vehicles and between vehicles and the RSUs is decreased.

Finally, we present the effect of using the centralized RTDB in the Control Center and the QoD on the message loss ratio.

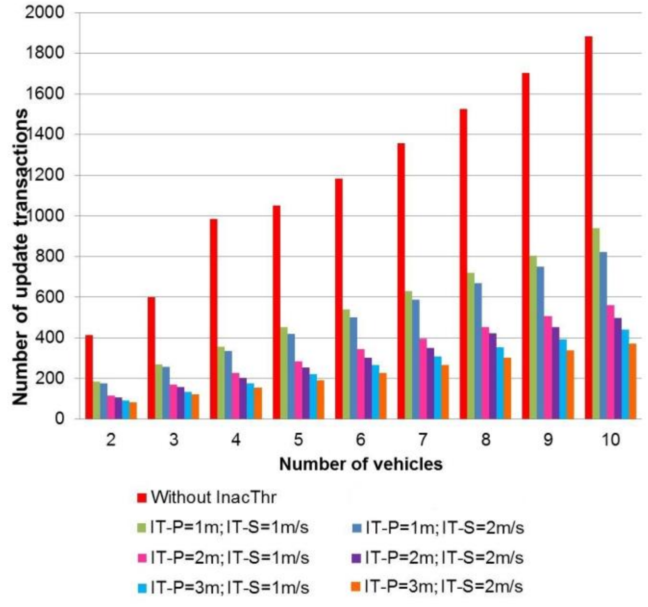

Figure 10. Comparison of the numbers of total update transactions, without InacThr versus with IT-S and IT-P in the predicted hazard scenario

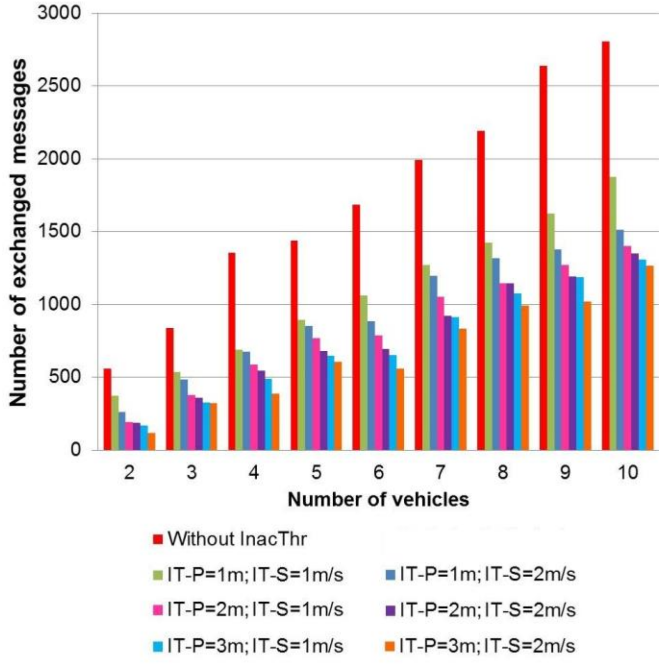

Figure 11. Comparison of the numbers of exchanged messages, without InacThr versus with IT-S and IT-P in the predicted hazard scenario

Figure 12 shows the variation of the loss message ratio without InacThr, and with IT-S equal to 2 m/s and varied IT-P values. For the case of ten running vehicles, this ratio is 80 % (red chart) without the use of IT-P and IT-S.
This ratio is decreased with the use of different values of the InacThr (the other charts): it is 56 % when IT-S is equal to 2 m/s and IT-P is equal to 3 m (orange chart), for the same case. We note that we have obtained a similar improvement with I-TS equal to 1 m/s. So, we can conclude that the InacThr for both position and speed decrease the number of lost messages, and therefore enhance the performance of the Predictable CopRoadHazDPS while assuring its robustness. Like the first scenario, this reduction is insufficient; it can be improved further by other means [33, 34].

3) *Recapitulation*

On the one hand, the simulations confirm (i) the robustness of CopRoadHazDPS as it avoids road accidents with unexpected or expected hazards, and (ii) its efficiency as it reacts imminently.
On the other hand, the aforementioned results show that the number of update transactions in the RTDB and the number of exchanged messages with V2V and V2I communications are reduced remarkably thanks to stored data.

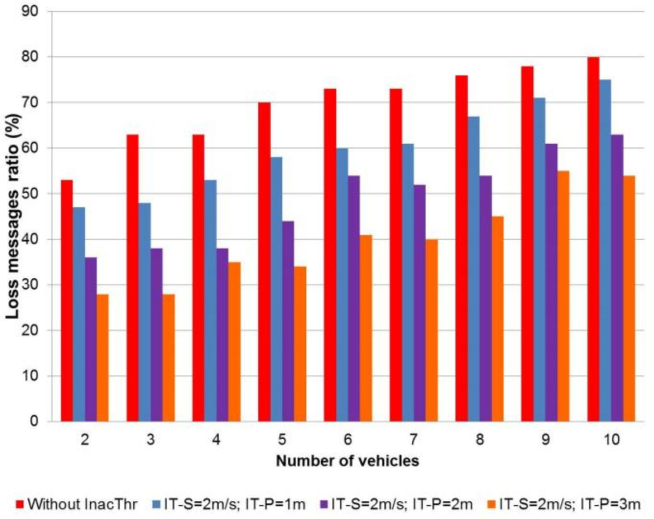

Figure 12. Variation of the ratio of loss messages, without InacThr versus with IT-S=2 m/s and varied IT-P in the predicted hazard scenario

On the other hand, the aforementioned results show that the number of update transactions in the RTDB and the number of exchanged messages with V2V and V2I communications are reduced remarkably thanks to stored data. Thus, the storage in an RTDB has a significant influence as the number of *write* operations in the RTDB and the risk of conflict between transactions are reduced thanks to the QoD. Hence, the RTDBMS achieves the expected performance by (i) handling a huge amount of data with easy and simple calculations, (ii) decreasing the exchanged information, and (iii) reducing the quantity of lost messages. In [8], the reduction of the numbers of update transactions leads to a reduction in the computing time. Likewise, all these reductions (*i.e.*, in terms of the numbers of transactions, exchanged and lost messages) have an impact on the rapidity of CopRoadHazDPS. Indeed, CopRoadHazDPS increase the road traffic throughputs as vehicles are able to reach their destinations quickly.
Consequently, we conclude that using stored data in an RTDB system along, with VANET communications, enhances the performance of a cooperative road hazard detection system appreciably.

VI. Conclusion

Unexpected and expected road dangers constitute a serious problem on roads, causing deaths, injuries, and material destruction. To prevent collisions with road hazards, we introduced in this paper an integrated system covering the two challenging factors: unpredictable and predictable road hazards. The proposed system, called *Cooperative Road Hazard Detection Persistent System* (CopRoadHazDPS), relies on VANET communications and RTDBMS. The first sub-system, Unpredictable CopRoadHazDPS, deals with the unexpected road hazards like static and dynamics obstacle, alert about accidents, etc. The second sub-system, Predictable CopRoadHazDPS, examines the expected road hazards in turn maneuvers in a T-intersections and on-ramp merging in highways. CopRoadHazDPS assists drivers in their trajectory and automatically takes the appropriate actions quickly when a road hazard is detected. It simplifies and accelerates the calculation operations as it uses stored data and the concept of QoD. The results of the simulations, realized under VEINS framework, confirm that the integration of RTDB, whatever in the vehicles or in the Control Center, improves performance in the context of C-ADAS.

In our future work, we intend to improve CopRoadHazDPS performance, on the one hand, and implement a Temporal DataBase in the Control Center to handle historical data, on the other hand. So, it will become possible to develop data mining processes about road hazards and their consequences, in order to make good decisions enhancing the road traffic safety.

References

[1] Association for safe international road travel, http://www.asirt.org/, last accessed 2018/09/20.

[2] A. Koesdwiady, R. Soua, F. Karray, M. Kamel. “Recent Trends in Driver Safety Monitoring Systems: State of the Art and Challenges”, *IEEE Transactions on Vehicular Technology*, 66 (4), pp. 4550–4563, 2016.

[3] C. M. Martinez, M. Heucke, F. Y. Wang, B. Gao, D. Cao. “Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey”, *IEEE Transactions on Intelligent Transportation Systems*, 19(3), pp. 666-676, 2018.

[4] S. E. Shladover. “Connected and automated vehicle systems: Introduction and overview”, *Journal of Intelligent Transportation Systems*, 22(3), pp. 190-200, 2018

[5] A. Dahiya, R.K. Chauhan. “Comparative study of MANET and VANET Environment”, *Journal of Computing*, 2 (7), pp. 87-92, 2010.

[6] A. A. Taleb. “VANET Routing Protocols and Architectures: An Overview”, *Journal of Computer Science*, 14 (3), pp. 423-434, 2018.

[7] R. I. Meneguette, E. Robson, A. A. Loureiro. “Vehicle-to-Infrastructure Communication”, *In Intelligent Transport System in Smart Cities*, pp. 53-77, 2018.

[8] I. Elleuch, A. Makni, R. Bouaziz. “Design of an Intelligent Cooperative Road Hazard Detection Persistent System”, *In Proceedings of the International Conference on Intelligent Systems Design and Applications (ISDA)*, pp. 696-707, 2018.

[9] C. Pang, X. Zhong, H. Huosheng, J. Tian, P. Xiafu, Z. Jianping. “Adaptive obstacle detection for mobile robots in urban environments using downward-looking 2D LiDAR”, *Sensors journal*, 18(6), pp. 1-18, 2018.

[10] F. Castaño, G. Beruvides, R. Haber, A. Artuñedo. “Obstacle recognition based on machine learning for on-chip LiDAR sensors in a Cyber-Physical System”, *Sensors journal*, 17(9), pp. 2109-2123, 2017.

[11] S. Yenkanchi, Q.M. Jonathan. Cooperative fusion for road obstacles detection using laser scanner and camera. In Intelligent Control and Automation (WCICA), *12th World Congress on IEEE*, pp. 983-986, 2016.

[12] M. N. Mariyasagayam, H. Menouar, M. Len. “GeoNet: A project enabling active safety and IPv6 vehicular applications”, *In proceedings of IEEE International Conference on Vehicular Electronics and Safety*, pp. 312-316, 2008.

[13] P. Gomes, C. laverri-Monreal, F. Michel. “Making vehicles transparent through V2V video streaming”, *IEEE Transactions on Intelligent Transportation Systems*, 13(2), pp. 930-938, 2012.

[14] S. Toth, J. Janech, E. Krák, “Query Based Image Processing in the VANET”, *In Computational Intelligence, Communication Systems and Networks (CICSyN), 5th International Conference on IEEE*, pp. 256-260, 2013.

[15] B. Sathyasri, P. P. Alphons. “Accident alert message transmission using VANET”, *International Journal of Pure and Applied Mathematics*, 118(20), pp. 3327-3334, 2018.

[16] H. Najada, I. Mahgoub. “Anticipation and alert system of congestion and accidents in VANET using Big Data analysis for Intelligent Transportation Systems”, *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on IEEE*, pp. 1-18, 2016.

[17] H. El-Sersy, A. El-Sayed. “Survey of Traffic Congestion Detection using VANET”, *Foundation of Computer Science (FCS)*, 1(4), pp. 14-20, 2015.

[18] M. M. Dongre, G. B. Narendra. “Effective Road Model for Congestion Control in VANETs”, *International Journal of Wireless & Mobile Networks (IJWMN)*, 8 (2), pp. 11- 21, 2016.

[19] HERE Hazard Warnings, https://www.here.com, last accessed 2019/02/18.

[20] M. Schmitz, L. Rittger, H. Kienast, A. Neukum. “HMI for Left Turn Assist (LTA)”, *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, 11(07-11-01-0002), pp. 16-29, 2018.

[21] P. Wang, S. E. Fang, L. Zhang, J. Wang. “A Vehicle Collision Detection Algorithm at T-Shaped Intersections Based on Location-Based Service”, *In New Frontiers in Road and Airport Engineering*, pp. 308-317, 2015.

[22] Y. Rahmati, A. Talebpour. “Towards a collaborative connected, automated driving environment: A game theory based decision framework for unprotected left turn maneuvers”, *In 2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1316-1321, 2017.

[23] I. A. Ntousakis, I. K. Nikolos, M. Papageorgiou. “Optimal vehicle trajectory planning in the context of cooperative merging on highways”, *Transportation*

research part C: emerging technologies, 71 (2016), pp. 464-488, 2016.

[24] J. Rios-Torres, A. A. Malikopoulos. "Automated and cooperative vehicle merging at highway on-ramps", *IEEE Transactions on Intelligent Transportation Systems*, 18(4), pp. 780-789, 2017.

[25] P. Wang, C. Y. Chan. "Autonomous Ramp Merge Maneuver Based on Reinforcement Learning with Continuous Action Space", *arXiv preprint arXiv:1803.09203*, 2018.

[26] Y. J. Li. "An Overview of the DSRC/WAVE Technology", *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, pp. 544–558, 2010.

[27] I. Elleuch, A. Makni, R. Bouaziz. "Cooperative Intersection Collision Avoidance Persistent System Based on V2V Communication and Real-Time Databases", *In the proceedings* of *IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1082-1089, 2016.

[28] The Tunisian Ministry of Transport: Code de la route. p. 544, 2008.

[29] C. Sommer, R. German, F. Dressler. "Bidirectionally coupled network and road simulation for improved IVC analysis", *IEEE Transactions on Mobile Computing*, 10(1), pp. 3-15, 2011.

[30] M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz. "SUMO – Simulation of Urban Mobility", *In the Proceedings of International Conference on Advances in System Simulation (SIMUL)*, pp.55-60, 2011.

[31] A. Varga, R. Hornig. "An Overview of the OMNeT++ Simulation Environment", *In the Proceedings of the First International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems & workshops (SIMUTOOLS)*, pp. 1-10, 2008.

[32] X. Ma, I. Andréasson. "Driver reaction delay estimation from real data and its application in gm-type model evaluation", *Transportation Research Record*, no. 1965, pp. 130-141, 2006.

[33] H. Wymeersch, G. R. de Campos, P. Falcone, L. Svensson, E. G. Ström. "Challenges for cooperative ITS: Improving road safety through the integration of wireless communications, control, and positioning", *In the proceedings of International Conference on Computing, Networking and Communications (ICNC)*, pp. 573-578, 2015.

[34] W. Lai, W. Ni, H. Wang, R. P. Liu. "Analysis of average packet loss rate in multi-hop broadcast for VANETs", *IEEE Communications Letters*, 22(1), pp. 157-160, 2018.

Author Biographies

Islam Elleuch received her Engineering degree in Computer Science from the National Engineering School of Sfax, Tunisia in 2014. She is currently a PhD student at Sfax University, Tunisia and a member of Multimedia, InfoRmation systems and Advanced Computing Laboratory (MIRACL). Her current research interests include intelligent transportation systems, cooperative advanced driver assistance systems, inter-vehicle communication and real-time databases.

Achraf Makni is associate professor on computer science at the National School of Electronics and Telecommunications of Sfax, Sfax University, Tunisia. His PhD has dealt with concurrency control in temporal databases. Currently, his main research topics of interest are temporal databases, real-time databases and intelligent transportation systems.

Rafik Bouaziz is full professor on computer science at the Faculty of Economic Sciences and Management of Sfax University, Tunisia. He was the president of this University during August 2014 – December 2017, and the director of its doctoral school of economy, management and computer science during December 2011 – July 2014. His PhD has dealt with temporal data management and historical record of data in Information Systems. The subject of his accreditation to supervise research was "A contribution for the control of versioning of data and schema in advanced information systems". Currently, his main research topics of interest are temporal databases, real-time databases, information systems engineering, ontologies, data warehousing and workflows.