Experimentation of a road hazard anticipation system based on vehicle cooperation

Bertrand Ducourthial, Véronique Cherfaoui, Thomas Fuhrmann, Stéphane Bonnet

Vehicular Communications 36 (2022) 100486

Contents lists available at ScienceDirect

Vehicular Communications

www.elsevier.com/locate/vehcom

Experimentation of a road hazard anticipation system based on vehicle cooperation

B. Ducourthial *, V. Cherfaoui, T. Fuhrmann, S. Bonnet

Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60319 - 60203 Compiègne Cedex, France

ARTICLE INFO

Article history:
Received 29 November 2021
Received in revised form 8 April 2022
Accepted 10 May 2022
Available online 16 May 2022

Keywords:
Road safety
Cooperative vehicle safety systems
Vehicular networks
Distributed data fusion
Conditional transmissions
Road experiments

ABSTRACT

This paper presents a complete system for anticipating road hazard based on inter-vehicular communication. It relies on efficient cooperative strategies between vehicles, namely *distributed data fusion* for hazard detection and *conditional transmissions* for alert propagation. It is built upon cheap embedded hardware and specific embedded software dedicated to dynamic vehicular networks. We describe the system, its architecture and the cooperative strategies. Road experiments involving 10 vehicles are reported. Real-life in-lab experiments complete the study with performance evaluation. These experiments emphasize the complementarity of both cooperative strategies. They show the practical interest of the whole system as well as its truthfulness and robustness even against poor networking conditions. They prove its efficiency to provide a significant anticipation period of road hazards.

1. Introduction

The design of hazard anticipation systems is motivated by road safety. Road fatalities have drastically decreased in the European Union (EU) during the last decades. Fatal crashes fell by 42% from 2000 to 2010 and by 23% from 2011 to 2019 [41,44]. This gives 51 deaths per million inhabitants in the EU (106 in the USA).

Nevertheless, such good figures should not hide a cruel and harsh reality. There is almost one million road accidents per year in the EU, leading to 1.2 million injuries (more than 3,000 per day) and 22,700 fatalities (more than 60 per day) [43]. Moreover, the figures seem to be stagnating and the EU objectives are not being reached (Fig. 1). The substantial decrease in 2020 is mainly explained by the exceptional circumstances due to the COVID-19 [44].

In this context, road anticipating systems are of great importance. Their aim is to give drivers more time to adapt their driving while approaching dangerous situations (for instance by slowing down). Though they could help in any situation, they apply mainly to non urban areas. Such areas represent 62% of the injuries in the EU (including 8% on motorways) [42]. Considering only the cars, the proportion reaches 77% (including 10.4% on motorways) [43].

The detection of road events (accidents, objects, extreme weather conditions...) is necessary to support road safety applications – and also traffic efficiency applications – by alerting road users of hazards. Road events are usually detected by infrastructure sensors, such as video surveillance cameras [60], by driver action using mobile applications or by messages sent by the car to a call-center in the event of an accident [5]. Advanced Driver Assistance Systems (ADAS) are developed for detecting hazards in the vicinity of the vehicles [24]. They rely on sensors included in new vehicles. Retrofitting such systems in older vehicles appears to be beneficial [50].

In the context of Intelligent Transport System (ITS), the next step would consist in the design of ADAS announcing road hazard to remote and approaching vehicles [36]. The deployment of such an approach is facilitated by standardization efforts related to vehicular communication [31,30,58]. Nevertheless, information has to be reliable and the propagation has to be adapted to the kind of alert (both topics addressed in this paper), in terms of area of distribution and duration among other criteria.

For this purpose, data collected by embedded sensors of closeby vehicles could contribute to a collaborative detection of road hazard, reinforcing the confidence into the events. Indeed, while a road event may be detected by a single vehicle thanks to its embedded sensors, in many situations it would be preferable to confirm it before warning remote vehicles. Reducing false positives will have strong consequences on the usability and efficiency of a remote road hazard alert system.

* Corresponding author.
E-mail address: Bertrand.Ducourthial@utc.fr (B. Ducourthial).

https://doi.org/10.1016/j.vehcom.2022.100486
2214-2096/

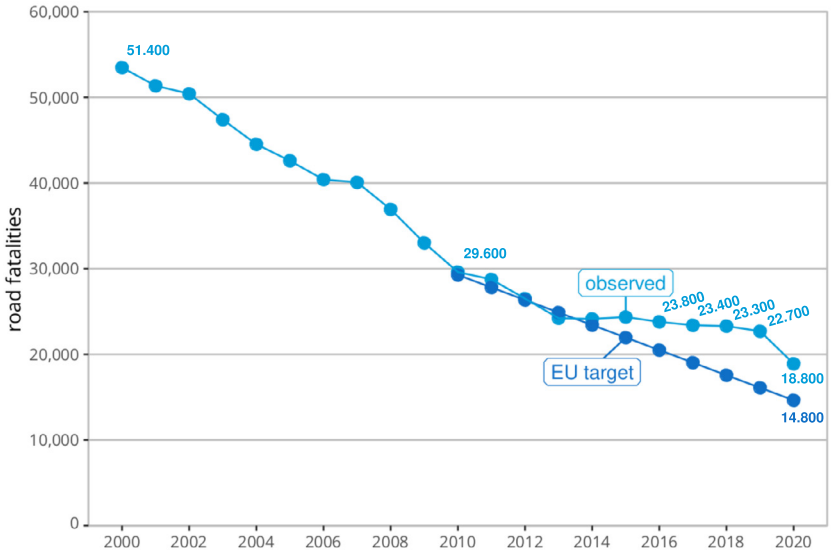

Fig. 1. Road fatalities in the EU (from [44] with figures from [43]).

Event confirmation can be achieved by centralizing collected data through a wireless network provider (e.g. 5G), sending and analyzing large amount of data on remote servers and sending back alerts to affected vehicles. This approach requires large computing and communication resources. Moreover it may delay alert propagation. Additionally, the data collection may lead to privacy issues. Another approach relies on cooperative distributed strategies where the alert decision is taken inside the vehicular network.

Finally, when the hazard is confirmed, the propagation of the road alert has to be done efficiently. The area of distribution should be alert-dependent. For instance, a weather alert may be propagated up to a distance depending on the event; an accident alert could be propagated only to approaching vehicles on the related lane. In general, the less involved non-concerned vehicles are, the more efficient the propagation becomes, in terms of network resources and driver attention. Yet, collaborative distributed strategies could contribute to efficient alert propagation into event-dependent area of distribution.

Hence, designing a road hazard anticipation system raises several challenges:

- reinforcing the confidence into the detected event,
- preserving the privacy of road users by limiting the amount of collected data,
- preserving bandwidth, communication and processing resources (thus minimizing energy consumption),
- minimizing the time to efficiently warn other drivers.

In this paper, we propose a general system for hazard anticipation in vehicular networks relying on cooperative hazard detection and cooperative alert propagation. The hazard detection is based on distributed data fusion to reinforce the confidence into an event, taking into account the uncertainties in both the data and the sources of information. The alert propagation is based on conditions to better fit with alerts characteristics while optimizing vehicle-to-vehicle communications.

The contribution is twofold. First, we present the design of a complete system for hazard anticipation, partially based on previous independent works [19,18,17]. Second we provide a proof of concept and an extensive study of the system performance showing its interest to reinforce confidence into events and efficiently warn affected vehicles.

For this purpose, the paper presents the global architecture and its data-flow. Then it presents an open-road, real-life experiment of the whole system, involving ten vehicles and cheap embedded hardware, showing the feasibility of our approach (and its compatibility with ADAS retrofitting [50]). These road experiments are completed by a performance study of the whole system by means of real-life in-lab experiments. These experiments demonstrate the complementarity of both cooperative strategies (hazard detection and alert propagation). They show the truthfulness in alert delivering and the robustness of the system against poor networking conditions. They point out a significant anticipation period to the drivers.

The rest of this paper is organized as follows. The next section summarizes the related work to position our contribution. Then, the architecture of our system is presented in Section 3. The cooperative hazard detection is presented in Section 4 and the cooperative alert propagation in Section 5. The open-road proof of concept is presented in Section 6. The performance evaluation is detailed in Section 7. Concluding remarks end the paper.

2. State of the art

Many studies related to intelligent transportation systems are devoted to road safety in the context of communicating vehicles. The eCall system is deployed in Europe to warn rescue teams in case of car accidents [5]. The alert is sent automatically in case of serious damage in the vehicle.[1] To anticipate and prevent accidents, safety warning systems aim to warn the driver before the road hazard [36]. The challenge is the reliability of the detection, avoiding both false positives and false negatives.

Cooperative intelligent transportation systems are a promising technology to increase the reliability of hazard detection. They rely on vehicle-to-vehicle communication [2]. In [64], an opportunistic approach is proposed to group vehicles in order to cooperatively detect road events. The data are aggregated inside the groups from the follower to the leader using for instance the average function. In [59], the beacons of many vehicles are exploited to detect traffic congestion using relationship between events and complex patterns recognition.

[1] A passenger may also manually send an alert using eCall.

Table 1
Related works synthesis.

Domain	References and years	Methods
Hazard detection	[5] (2017), [23] (2016), [24], [36] (2020)	Smartphone, eCall
	[22][59] (2012), [64] [36] 2010- 2020	Cooperative strategies
	[29] (2010) , [46] (2018), [62] (2020)	Confidence degree, reputation
Data fusion	[51] (1976), [40] (2007), [25] (2012)	Synthesis
	[35] (2013), [27,37] (2021), [63] (2022)	Filtering, estimation methods
	[48] (2008), [18,21] (2012), [39] (2016), [49] (2020)	Uncertainties management
Alert propagation	[16] (2014), [9] (2016), [8,28] (2017), [53] (2019), [33] (2021), [52] (2022)	Synthesis
	[30] (2013), [31] (2014), [32] (2017), [58] (2020)	Standard
	[6] (2000), [4] (2003), [32] (2017)	Rouging, geocast
	[45] (2018), [57] (2019), [3] (2021)	Clusters
	[19] (2007), [15] (2015), [61] (2020), [65] (2021), [26] (2022)	Cooperation, content centric

Nevertheless, cooperative detection may be polluted by malicious or faulty vehicles. In [29], some filters are applied so that a driver is only warned after several sources do agree. In [46], a trust degree is joined to the exchanged information (related to the map of closeby vehicles). The trust degree is updated during the forwarding process depending on the number of independent sources confirming the information. To avoid fake alerts generated by malicious nodes, [62] relies on the reputation of the vehicles in the so-called Vehicular Social Networks to select the next forwarder.

The data fusion aims at combining several sources of information to reduce the uncertainties and false alarms. Much work has been done on Bayesian inference [40]. The Dempster-Shafer theory [51] appears to be well adapted to manage the lack of knowledge and the conflicts between sources in the road event detection systems. In [48], the authors shows its interest for trust management in vehicular networks by comparing the Dempster rule to Bayesian inference and weighted voting.

In this context, the distributed data fusion methods avoid collecting the data before their combination [25]. In [35,27,37], local sensor data are combined with received data to construct a map of neighbors with a distributed Kalman filter or split covariance intersection filter. In [63], a cooperative localization scheme is used to perform better positioning system when the GNSS is not available. In [21], distributed algorithms are proposed to detect Sybil attacks. This work is extended in [39] to congestion detection. In [49], the authors propose a framework that fuses data from heterogeneous data sources to enhance Intelligent Transportation System services, such as vehicle routing and traffic event detection. In [18], a robust distributed data fusion algorithm is proposed; it supports transient failures and topology changes. An application is given in [47] for detecting icy road using sensors of the infrastructure combined with those embedded in approaching vehicles. The system has been tested on the road with three sensors and one vehicle. Thanks to its robustness, it supports misplaced sensors.

Such cooperative hazard detections are robust; they allow warning the driver to prevent accidents. However, to increase the anticipation period, the alert could be sent to approaching vehicles, not yet involved into the distributed data fusion. For instance in [23], the accidents are detected using either an OBD-II connector or smartphone sensors. Then the alert is propagated using an MSD[2] message sent using eCall or a DENM message [30] sent using IEEE 802.11p.

There exist mainly two families of algorithms that may be used to send an alert in the vehicular network: broadcast or multicast/geocast [16,9,53,52]. The first family tries to send a message to all nodes. The second one tries to select the target nodes. It is more interesting for alert propagation because it is able to limit information transmission to vehicles having a possibility to reach the hazard [6,4]. This may reduce the impact on the network resources and avoid disturbing non affected drivers. The ETSI published the specification of geonetworking protocols [32,53]. These protocols rely on geographical positions to forward messages to a single receiver or a group of receivers. Other protocols rely on groups of vehicles sharing some characteristics (clusters) [16,8,52]. In [45], a trust management scheme is used to determine the clusters. In [3], clusters are combined to Named Data Networking for routing messages according to data content. In [57], a cluster strategy improving OLSR is used for optimizing both the cooperative localization and the cooperative perception.

Additionally, cooperative strategies have been proposed to forward the messages in VANETs. Here, the forwarding process is more intricate with the requesting application or with the data to be forwarded in the aim of better facing the hard vehicular network conditions [16,15,28]. In [33], the authors study a carry-and-forward strategy exploiting the trajectories of vehicles in the aim of reaching Internet access points sparsely deployed along the roadways. In [61], a probability of interest is computed based on an estimation of the network density. In [28], a review of approaches promoting the cooperation in the vehicular network is proposed, including among other incentive-based, reputation-based or misbehavior-based approaches. In [65], cooperative transmissions with multiple relays are studied; the authors show the importance of node coordination. In [26], an emergency message dissemination strategy is proposed; it is based on the vehicle speed and on the collaboration of neighbors. With conditional transmissions [19], the forwarders and the recipients of the alert are determined thanks to some conditions evaluated at reception. The condition may be related to trajectories correlation for restricting the propagation to a given lane, to the distance and duration for limiting the dissemination, and so on. They are application-dependent and the technique is then related to data-centric routing.

Table 1 proposes a synthesis of the related work. Cooperative hazard detection systems relying on distributed data fusion appear to be promising. However up to now they have mainly been tested in simulation or small-scale experiments only involving 2 or 3 vehicles. Moreover in the aim of providing a significant anticipation period, it could be combined with a compatible alert propagation strategy. But to the best of our knowledge, there is no such study.

In this paper, we design and experiment a complete cooperative road hazard anticipation system combining both detection and prevention. Our design choices focus on the distributed data fusion algorithm presented in [18] (for hazard detection) and on the conditional transmissions [19] (for alert propagation). First, the distributed data fusion enforces the robustness of the hazard detection. By using the Dempster-Shafer approach, it is able to manage the lack of knowledge or the conflicts in the vehicular

[2] The Minimum Set of Data (MSD) is the data component of an eCall sent from a vehicle to an emergency call center. Its maximum size is 140 bytes; it may include vehicle identity, location information and time-stamp [1].

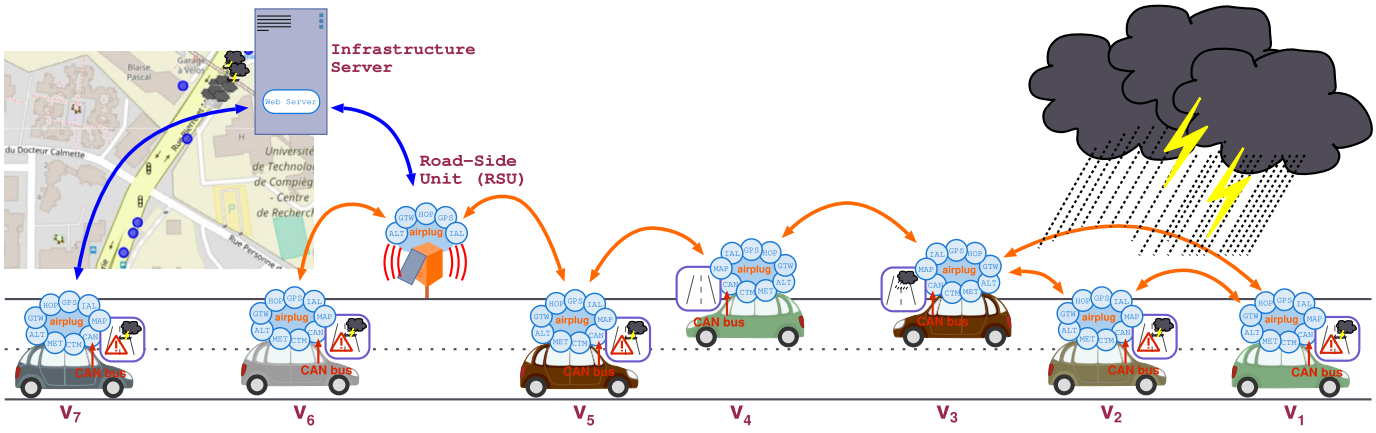

Fig. 2. Illustration of the cooperative hazard anticipation system. The hazard is detected by vehicles v_1, v_2 and v_3 using distributed data fusion. The alert is propagated by a selection of vehicles (v_3, v_4 and v_5) using conditional transmissions. The RSU forwards it to approaching vehicles (v_6) and also to the Internet for remote vehicles (v_7). Vehicle-to-vehicle communication is drawn in orange; others (in blue) may be done by a network operator. The anticipated alert is displayed to drivers approaching the road hazard.

network. Second, the conditional transmission approach is versatile enough to limit the number of involved vehicles regarding the type of road hazard. By relying on receiver-side decisions, it avoids collecting information about the neighborhood and is then more robust to network dynamics and packet loss. Hence both cooperative strategies are robust against topology changes or messages losses. We show the interest of combining both of them to provide a significant anticipation period.

We present road experiments involving ten vehicles and extensive in-lab experiments relying on network emulation. A particular attention is given to the truthfulness and robustness of the system, in particular against weak networking conditions.

3. Overall system description

This section presents the principle, the structural and the functional architecture of our cooperative hazard anticipation system.

3.1. Principle

The detection relies on a cooperation between nearby vehicles in such a way the local data of each vehicle contributes anonymously to the computation of a common information. Such common information is the basis of the hazard detection. When it is detected, an alert is propagated from vehicle to vehicle in a scheme ensuring that only affected vehicles are warned. The alert is also sent to road-side units (and to the Internet if required by the type of alert) to warn distant road users.

By combining a distributed algorithm for hazard detection and an optimized alert propagation protocol, network resources are used parsimoniously. Moreover, only the common information related to the alert – computed cooperatively – is sent remotely, thus preserving privacy.

Computing a common information reinforces the reliability of the system. Indeed, some sources of information may be erroneous or misplaced, generating false alerts. Our algorithm fuses the input data of involved vehicles, giving more importance to closer sources than farther ones. It relies on the Dempster-Shafer theory. Thanks to its properties, it converges rapidly despite the unknown and dynamic topology of the vehicular network.

To alert only the affected vehicles, our propagation protocol relies on *conditional transmissions*. The alert is propagated according to alert-dependent conditions related to the geographic area, the vehicles trajectories, the alert duration, etc. This protocol encompasses road-side units discovery by using suitable conditions.

Fig. 2 illustrates our cooperative hazard anticipation system. In this example, the hazard is detected by vehicles v_1, v_2 and v_3 (using distributed data fusion). It is propagated by vehicles v_3, v_4 and v_5 (using conditional transmissions). The road-side unit is in charge of forwarding the alert to approaching vehicles (v_6) and to the Internet for remote vehicles (v_7).

3.2. Components of the system

The architecture of our cooperative road hazard anticipation system can be subdivided into five parts.

1. Sensors. Vehicles contribute to the hazard detection with a data input giving information related to the hazard to be detected. This data mainly comes from sensors connected on the vehicle CAN[5] bus. It is converted periodically into a specific format that includes a confidence level (see Section 4.1).
2. On-board unit. Each vehicle is equipped with an embedded computer able to read the input devices (data source, GNSS[3], etc.) and to exchange messages with other vehicles or road-side units using IEEE 802.11 standards (Sec. 6.1).
3. Middleware. Each embedded computer runs a framework offering facilities for splitting the tasks into independent applications communicating by local or V2V[4] messages (*Airplug*, see Sec. 6.1).
4. Applications. The vehicles run a set of applications (Fig. 3-left) which can be gathered into 3 groups:
   4.1 The CAN,[5] CTM[6] and MET[7] applications implement the cooperative hazard detection strategy. They are respectively in charge of i) reading and decoding the CAN bus frames, ii) transforming the input data into a so-called *local confidence* in the feared event occurrence and iii) computing on-the-fly the so-called *distributed confidence* (Sec. 4). To this aim, MET implements a distributed algorithm (distributed data fusion) relying on V2V communications (orange arrows in Fig. 2).

[3] Global Navigation Satellite System.
[4] V2V stands for *Vehicle-to-Vehicle*.
[5] CAN stands for *Controller Area Network*.
[6] CTM stands for *CAN to MET*.
[7] MET stands for *meteorology* though the application is very generic and is able to compute the distributed data fusion for any use-cases (providing appropriate inputs are given by CTM).

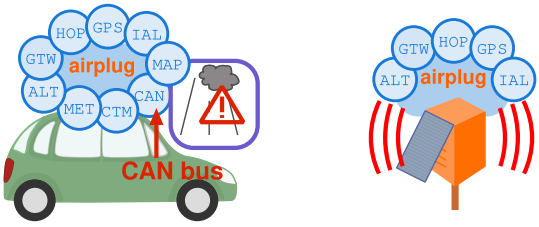

Fig. 3. Software applications of the hazard anticipation system used in the vehicles on the left and in the road-side units on the right. They run on top of a light message-passing framework named Airplug, in charge of both inter- and intra-vehicle communications.

4.2 Next, the ALT,[8] GTW[9] and HOP[10] applications implement the cooperative alert propagation strategy. They are respectively in charge of iv) generating and receiving alerts, v) discovering the Internet gateways and vi) propagating them intelligently (Sec. 5). To this aim, HOP implements a multi-hop communication protocol (conditional transmissions) relying on V2V and V2I[11] communications.

4.3 Finally, the GPS, IAL[12] and MAP applications are respectively in charge of vii) providing the geographic vehicle position to interested applications, viii) discovering neighbors, and ix) displaying neighbors and alerts to the driver. The neighbors discovery is actually optional for the hazard anticipation system but more convenient in practice (see maps in Fig. 9).

5. Infrastructure. The last part of our system is infrastructure-related. Road-side units (RSU) record the alert messages, replay them to arriving vehicles and publish them to a web server (Fig. 2). As seen in Fig. 3-right, an RSU only embeds the applications related to the alert propagation (ALT, GTW, HOP) and (optionally) to neighbors discovery (GPS, IAL) to be displayed on the maps of approaching vehicles (see Fig. 9).

3.3. Functional architecture

The functional architecture of the system is presented in Fig. 4. It can be read from left to right for the ego vehicle (from the hardware to the driver) though it includes inter-vehicle communication (top-bottom reading).

The inputs of the system are given by the embedded sensors connected to the CAN[5] bus of vehicles. Local applications are grouped in grey boxes; intra-vehicle communication is in red. Blue boxes represent distributed applications relying on inter-vehicle communication (in orange). The final output of the system consists in an anticipated alert displayed to the approaching driver.

The distributed algorithms involve vehicles in bounded areas around the ego vehicle. The nested blue boxes are related to the size of such areas in the vehicular network. Note that the area related to cooperative hazard detection and cooperative alert propagation can be configured thanks to some parameters of the algorithms and the smaller one is not necessarily included into the larger one. For instance the hazard detection may involve any vehicle up to three hops from the ego vehicle according to the *discounting parameter* (Sec 4.3) while the alert propagation may involve only approaching vehicles up to ten hops on a given lane thanks to a specific *forward condition* (Sec 5.2).

[8] ALT stands for *Alert*.
[9] GTW stands for *Gateway*.
[10] HOP stands for *Multihops* communication; it implements the conditional transmissions.
[11] V2I stands for *Vehicle-to-Infrastructure*.
[12] IAL stands for *Is alive*.

The functional architecture can be subdivided into five parts (see numbers in Fig. 4):

1. Inputs are provided by the embedded devices.
2. The neighbors discovery algorithm involves inter-vehicle communication in the neighborhood of the ego vehicle (one hop); see the inner blue box in Fig. 4.
3. The cooperative hazard detection (based on distributed data fusion, Sec. 4) relies on inter-vehicle communication to a few hops. The algorithm integrates a mechanism for bounding the number of hops around the vehicle (*discounting*, see Section 4.3).
4. The cooperative alert propagation (based on conditional transmissions, Sec. 5) requires generally more hops to warn approaching vehicles. It may involve RSU to warn far vehicles depending on the kind of alert (case of vehicles v_6 and v_7 on Fig. 2).
5. The map gathers information including road, neighbors and alerts to the intention of the driver. This is the output of the system.

The complete data-flow of the system is given in Appendix A.

4. Cooperative hazard detection

In this section, we explain how the road events are detected cooperatively by means of inter-vehicle communication. Fig. 5 illustrates how this subsystem works.

4.1. Input source of data

Our cooperative hazard detection system relies on inputs produced periodically by the vehicles (e.g. embedded sensors or calculators). It is general enough to handle almost any kind of inputs, provided they can be interpreted as a confidence in the occurrence of the event to be detected.

Nevertheless, for our experiments (Sections 6 and 7), we use the windscreen wipers speed for three reasons: i) this information is related to some feared weather conditions; ii) this is a representative case of data that should be reinforced and iii) it is convenient for experiments. Indeed, a large windscreen wipers speed can be interpreted as heavy rain that may be problematic for drivers. However, if such a large speed is observed on a single vehicle, this may be due to a driver cleaning the windscreen. Hence, it is mandatory to confirm such information before broadcasting an alert, making it a good use-case for a collaborative approach. Finally, operating the windscreen wipers during road tests is easy and can be done even in fair weather. This greatly simplifies the road experiments compared to using actual rain detectors equipping some vehicles.

In general, data from embedded sensors can be read through the CAN bus installed on all recent vehicles. Unfortunately, the CAN bus frames encoding are generally proprietary on cars. Thus, a preliminary stage consists in understanding and decoding the required frames for every vehicle model involved in the experiment. Next, the CAN application has to be configured with respect to the seeked information (here the windscreen wipers speed) and the vehicle model it is running on.

4.2. Local confidence

The next step consists in converting the input data into a *local confidence* usable by our distributed data fusion algorithm. It represents the confidence of the vehicle into the occurrence of the feared event. This is the contribution of the vehicle to the cooperative detection. It is built from the sensor measurement (here the

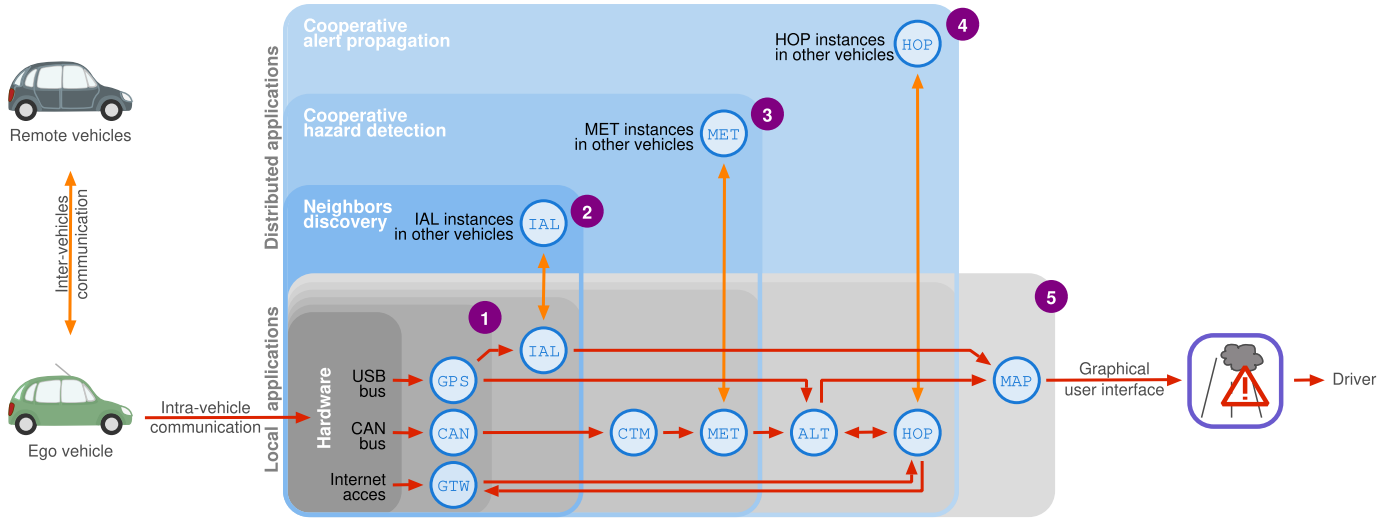

Fig. 4. Functional architecture of the cooperative hazard anticipation system. Local communication is drawn in red while inter-vehicle communication is drawn in orange. Local applications are in grey boxes while distributed applications are in blue boxes. The input of the system is given by the embedded sensors of some vehicles (ego or remote vehicles); the output is an anticipated alert displayed to the driver approaching the feared event.

windscreen wiper speed) and the confidence into this value, which is correlated to the confidence into the input device. Then, starting from the local confidences – periodically updated on each vehicle – our distributed data fusion algorithm computes the *distributed confidence* that will be used to detect the road hazard. Both the local and the distributed confidences use the same format, that we explain now.

Our distributed data fusion algorithm relies on the Dempster-Shafer Theory [18]. Appendix B presents a summary of the theory. This is a formal framework for reasoning with uncertainty that is well suited to manage unreliable information and weak states of knowledge [51,10]. It offers a diversity of combination operators. The incertitude is represented by means of so-called *belief functions*, that model the belief of a participant with respect to a set Ω named *frame of discernment*. A belief function can be represented (among other means) by *masses*, corresponding to values between 0 and 1 affected to the subsets of Ω so that their sum is 1.

Our algorithm (and its implementation in the Airplug MET application) is independent of the chosen frame of discernment. But to illustrate the idea, the following set has been used for the rain detection proof of concept:

$$\Omega = \{nofall, lowfall, highfall\} \tag{1}$$

For instance, a vehicle detecting an heavy rain could have a local confidence consisting in 0.8 on the subset $\{highfall\}$ and 0.2 on Ω (representing the doubt), supposing that our measurement device admits 20% of errors. The corresponding mass function (named m_1) is $m_1(\{highfall\}) = 0.8$ and $m_1(\Omega) = 0.2$ (the values on the other subsets are null). If the vehicle is less affirmative, the mass function (named m_2) could be: $m_2(\{lowfall, highfall\}) = 0.3$, $m_2(\{highfall\}) = 0.5$ and $m_2(\Omega) = 0.2$. More generally, a mass function can be written as a vector of $2^{|\Omega|}$ components, in our case 8 components (see Table 2).

When performing on-the-fly data fusion on a network with an unknown topology (as it is the case in vehicular networks), the main issue is the *data-incest problem* where a single source of information could be used multiple times [38,40]. For instance in Fig. 2, Vehicle v_1 includes its own local confidence into its computation of the distributed confidence, which is sent to its neighbors. When v_2 receives it, it includes it into its own distributed confidence computation, and sends the result to its neighbors. Then, v_1 will compute its new distributed confidence, taking into account its local confidence and the distributed confidence of v_2, which includes the previous distributed confidence of v_1. Hence, the local confidence of v_1 would be combined more than once into the computation.

To solve this problem, we use an idempotent operator [13], named the *cautious operator*. This operator is based on the *Least Commitment Principle*, which states that: "when several belief functions are compatible with a set of constraints, the least informative one should be selected". When the masses are converted into a vector of *weights* [54], the cautious operator is easily computed by taking the component-wise minimum. We then adopt such a format for representing both local and distributed confidences. The conversion from masses to weights of a vector of $2^{|\Omega|}$ masses gives a vector of $2^{|\Omega|} - 1$ weights (there is no component for Ω). The weights are real numbers belonging to $(0, 1]$. The conversion is detailed in Appendix C. Table 3 shows the vectors of weights w_1 and w_2 corresponding to masses m_1 and m_2 respectively.

To summarize, starting from the measurement produced by an embedded sensor or calculator (step 1 in Fig. 5) and retrieved from the CAN bus by the CAN application (step 2 in Fig. 5), the CTM application builds a vector of masses on the frame of discernment Ω (step 3 in Fig. 5). This vector is then converted into a vector of $2^{|\Omega|} - 1$ weights in the aim of optimizing the distributed computation. It represents the *local confidence* of the ego vehicle. It is used as input by the MET application, implementing the distributed computation, that we detail hereafter (step 4 in Fig. 5). The size of each component of such a vector depends on the discretization of the weights. To illustrate the idea, we discretize the weights up to ten thousandth in our proof of concept, leading to 32 bytes per vector.

4.3. Distributed confidence

Starting from its local confidence (regularly updated) and the last received messages, each vehicle computes periodically a *distributed confidence*, which is broadcast in the neighborhood.

This local computation combines – using the cautious operator – the local confidence of the vehicle and all the distributed confidences received from the neighbors since the last local computation. The distributed algorithm is asynchronous, meaning that the involved nodes do not need to synchronize their local computation. The local period should be adjusted in order to not saturate the bandwidth. In practice, we used a period of 1 second. It could

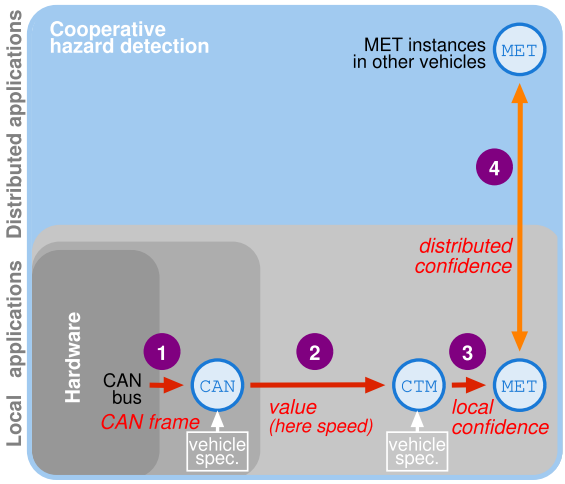

Fig. 5. Data exchanges during the cooperative hazard detection. The CAN application receives frames from the CAN bus (1) and extracts the windscreen wipers speed according to the vehicle model (2). The CTM application transforms the received speed into a vehicle-independent local confidence; this is the input for MET (3). The MET instances compute and exchange distributed confidence (4).

be adjusted depending on the dynamic of the event and the bandwidth occupancy.

By using the weights format, the local computation is a simple component-wise minimum on the vectors. However, each received confidence – coded as a vector of weights – is *discounted* before this computation so that it is less informative. We used the following discounting function during our experiments:

$$w \mapsto \min(1, w + 0.1) \tag{2}$$

By applying a discounting, the local confidences of distant vehicles become less important than closer ones, thus leading to an interesting property: while each vehicle contributes to the distributed computation, the result may differ from vehicle to vehicle. For instance, when approaching a dangerous area, the distributed confidence into the event increases (see Fig. 11).

As studied in [17], the discounting calibration impacts both the size of the area of influence of a given vehicle and the convergence time of the algorithm. With a small discounting, it is possible to influence vehicles in a larger area than with a large discounting. On the other hand, a large discounting ensures a faster convergence of the distributed algorithm. Hence, the discounting should be tuned according to the tracked event. Indeed, some events are large (e.g. weather alert) and it could be beneficial to involve distant vehicles. Conversely, some events are more dynamic and local (e.g. traffic slowdowns) and should not involve distant vehicles.

The discounting also ensures the self-stabilizing property of the algorithm [18], which means that it is able to converge in finite time even in case of transient failures. This property is important because it is nearly impossible to predict the set of cooperating vehicles nor the topology of the communication network. Thus the distributed algorithm has to support the dynamics of the vehicular network. For instance, in case a vehicle passes through a convoy of a few vehicles, the network topology is changed but this should not affect the distributed computation. Moreover, when a vehicle gets closer to the others with some very different inputs, the algorithm should still converge. This may happen for instance when a vehicle was into a tunnel and arrive meets others under an heavy rain; while the input in the arriving vehicle is not up-to-date, the disturbance of the distributed computation should be limited in time and amplitude.

To summarize, the distributed confidence is periodically computed by each node using a component-wide minimum on the local confidence and the received vectors of weights. Such received vectors are discounted in order to:

- Bound the area of influence of a vehicle (from discounting to discounting, its information becomes vacuous after a few hops, at most 10 in our case, see Eq. (2));
- Give more importance to the information of a nearby vehicle rather than those of a farther vehicle (this ensures the lack of false negatives, see Sec. 7.3);
- Ensuring the self-stabilizing property, which is mandatory for the convergence of the distributed application despite the dynamics of the vehicular network.

Table 2
Examples of masses on the frame of discernment Ω defined in Eq. (1).

	∅	{nofall}	{lowfall}	{highfall}	{nofall, lowfall}	{nofall, highfall}	{lowfall, highfall}	Ω
m_1	0	0	0	0.8	0	0	0	0.2
m_2	0	0	0	0.5	0	0	0.3	0.2

Table 3
Examples of weights obtained from masses m_1 and m_2 defined in Table 2.

	∅	{nofall}	{lowfall}	{highfall}	{nofall, lowfall}	{nofall, highfall}	{lowfall, highfall}	Ω
w_1	1	1	1	0.2	1	1	1	-
w_2	1	1	1	0.5	1	1	0.4	-

5. Cooperative alert propagation

In this section, we explain how the alerts are propagated cooperatively to affected vehicles. Fig. 6 illustrates how this subsystem works.

5.1. Alert emission

In order to take a decision, the vector of weights has to be mapped into an ordered set for comparison with some thresholds. There exist several methods for this purpose such as plausibility, credibility or pignistic probability [55]. The last one is commonly used because it enables some comparisons with probabilistic models; it is used in our system.

Starting from the distributed confidences, each node periodically computes the *pignistic probabilities* in order to make a decision [55]. For this purpose, the vector of $2^{|\Omega|} - 1$ weights is converted back into masses on the $2^{|\Omega|}$ subsets of Ω (see Appendix C), from which the probabilities into the $|\Omega|$ events of Ω are computed (in our proof of concept, probability of *nofall*, *lowfall* and *highfall*, see Eq. (1)). Consider $a \in \Omega$ and $A = \{a\}$ one of this event; let $m(B)$ the mass on $B \subset \Omega$. Then the pignistic probability $P_m(A)$ computed with mass m is given by:

$$P_m(A) = \sum_{\emptyset \neq B \subseteq \Omega} m(B) \frac{|A \cap B|}{|B|} \tag{3}$$

Table 4 details the computation of the pignistic probability for the event *highfall* from the masses m_1 and m_2 previously introduced in Table 2. The probability computed with m_2 is smaller than the one computed with m_1 because it is less affirmative, as explained in Sec. 4.2.

Table 4
Example of computation of the pignistic probability for the feared event $A = \{highfall\}$ with masses m_1 and m_2 defined in Table 2.

B	$\emptyset$	{nofall}	{lowfall}	{highfall}	{nofall, lowfall}	{nofall, highfall}	{lowfall, highfall}	Ω
m_1	0	0	0	0.8	0	0	0	0.2
$\frac{\lvert A\cap B\rvert}{\lvert B\rvert}$	0	0	0	1	0	0	0	$\frac{1}{3}$
						$P_{m_1}(A) = 0.8 \times 1 + 0.2 \times \frac{1}{3} \simeq 0.866$		
m_2	0	0	0	0.5	0	0	0.3	0.2
$\frac{\lvert A\cap B\rvert}{\lvert B\rvert}$	0	0	0	1	0	0	$\frac{1}{2}$	$\frac{1}{3}$
						$P_{m_2}(A) = 0.5 \times 1 + 0.3 \times \frac{1}{2} + 0.2 \times \frac{1}{3} \simeq 0.716$		

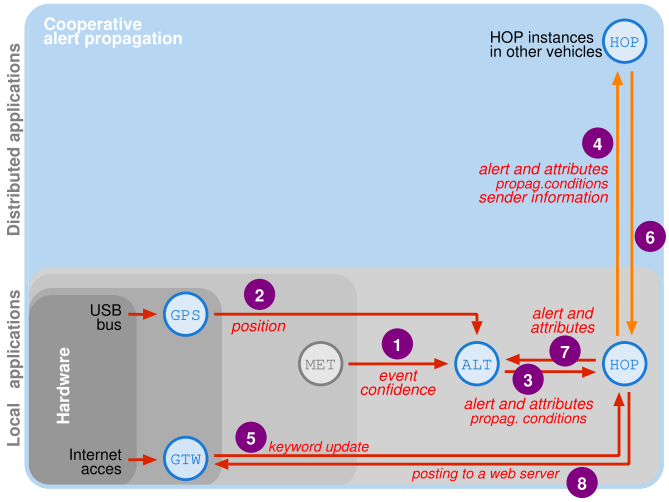

Fig. 6. Data exchanges during the cooperative alert propagation. When the alert sending threshold T_{snd} is exceeded, MET warns ALT about the feared event and its confidence (1). ALT generates an alert with its attributes, including confidence and position (2). It warns the ego driver and requests HOP (3) to reach remote drivers with specific propagation conditions. HOP sends to neighbors the alert, the propagation conditions and additional information for their evaluation at reception (4). When GTW discovers an Internet access, keywords values used by HOP in conditions evaluation are updated (5). At message reception (6), if the forward condition is true, HOP relays the message to other HOP instances in its vicinity (4). If the upward condition is true, it sends the message to the local ALT instance (7), to warn the ego driver. If an Internet access has been detected, the alert may also be posted into a web server (8).

These probabilities are not sent to the neighbors because our robust algorithm is based on richer information (confidences coded as vectors of weights, as seen in the previous section). They are used to generate the alert: when the probability into the feared event (here heavy rainfall) exceeds the *alert sending threshold* T_{snd}, an alert is generated.[13]

Then vehicles cooperate to propagate it towards affected vehicles.

For this purpose, the MET application periodically updates the pignistic probability into the feared event. When it exceeds T_{snd}, it warns ALT. The ALT application is in charge of generating the alert and/or exploiting the received alerts. The alert message includes the event type and some attributes such as the confidence into this event, a timestamp and the GPS position it was generated at. It also includes some event-dependent conditions defining the affected vehicles, the alert duration and the propagation.

[13] Besides the alert sending threshold T_{snd}, we also consider a pre-alert threshold T_{pre} to warn the ego driver only. We have $T_{\text{pre}} \leq 0.5$ and $T_{\text{snd}} \geq 0.5$; see Section 7.

When ALT generates or receives an alert, it warns the MAP application to add the related alert icon on the map to the intention of the driver (Fig. 4). The ALT application is able to periodically broadcast the alert message to neighbor vehicles, using some heuristics to avoid message collisions or message losses. However, a finer propagation scheme is preferable. Hence, ALT delegates the alert propagation to the HOP application, in charge of the cooperative alert propagation.

5.2. Cooperative propagation

Our cooperative alert propagation relies on conditional transmissions [19]. This is an efficient content-based diffusion working as follows. The alert sent to neighbors includes two conditions, the *forward* and the *upward* conditions. At reception, if the forward condition is true, the message is forwarded to the neighbors. Moreover, if the upward condition is true, it is forwarded to the upper layers, that is the local applications. Hence, a vehicle could be a relay, a receiver or both. The conditional transmissions are well adapted to dynamic networks because a vehicle v does not need to save information about its neighbors in the aim of determining which is the next hop. In some situation, such information could be costly to collect and unstable. Here, the decision to be a relay or not is taken by any node receiving the message sent by v (receiver-side decision). The forward and upward conditions are application-dependent. In our case, they depend on the type of alert and are provided by the ALT application.

Let's take the example illustrated on Fig. 2 where the alert type is *highfall rain*. The vehicles v_1, v_2 and v_3 experience heavy rain. Soon, on v_1 or v_2, the pignistic probability computed from the distributed confidence exceeds the threshold and an alert is generated. It is broadcast in the communication range of the vehicle with a forward condition of "being less than 4 km away" and an upward condition of "being on the same trajectory". When v_3 receives such a message, it forwards it (distance is less than 4 km) but does not warn its driver because it is going in the opposite direction (forward condition true, upward condition false). Vehicle v_4 acts similarly. When v_5 receives this message, both conditions are true: the alert is forwarded and also displayed to the driver. Conditional transmissions can use logical expressions related to time, distance, trajectories, type of node, etc. [19].

The HOP application is in charge of this multihop cooperative propagation scheme. On the vehicle that generates the alert, the HOP instance receives the alert message and the appropriate conditions from the local ALT instance. On a receiving vehicle, when the upward condition is true, the HOP instance forwards the alert message to the local ALT instance, which in turns warns the MAP application that displays it to the driver (Fig. 4).

The conditions depend on the type of alert to be propagated. Some of them concern only a given lane while some others may

Fig. 7. Setup for the proof of concept. Top-left: IEEE 802.11p modem and roof antenna for vehicles. Top-center: Raspberry Pi running Linux, the Airplug middleware and all the applications. Top-right: IEEE 802.11p road-side-unit installed on the roof of the laboratory, close to the road. Bottom: fleet of 10 vehicles used during the road tests.

concern a circular area; some of them are very localized while some others should be propagated at several kilometers; some of them are relatively brief while some others may have a long duration. Depending on the conditions, HOP adds to the message the information required to evaluate the conditions at the reception (such as the local position for the trajectory matching condition for instance). Note that usual broadcast algorithms including geocast cannot select vehicles appropriately in contrast to conditional transmissions.

5.3. Communication to the infrastructure

Conditions can also be used to send the alerts towards the Internet [34]. Indeed, if the upward condition is "being on the same trajectory or having an Internet access", it will be true on any node that detected a gateway to the Internet.

In our example (Fig. 2), the road-side unit is connected to the Internet. When the message sent by v_5 reaches the RSU, it is forwarded to the local application. Then the alert is published on a web server and also replayed to the arriving vehicle v_6 (which is out of the communication range of v_5). The distant vehicle v_7 may learn about the road hazard using a web client.

The GTW application is in charge of detecting the gateways towards Internet. When it detects such a gateway, it warns HOP and indicates the type of network, such as WiFi, Ethernet, 3G/4G... (Fig. 6). The related keyword has now the value "true" in the conditions evaluated by HOP. Gateways related conditions can then be refined to select the type of network to be used. In general, the keyword ANYNET (standing for "any network") is selected for an alert, though it is not the case for other non priority traffic. Note that the combination of GTW and HOP with appropriate conditions yields a cooperative sharing of the RSU, extending their communication range [34].

6. Experimental validation

In this section, we summarize the road experiments.

6.1. Setup

The base setup (named *Airbox*) is composed of a Raspberry Pi 3 board (Fig. 7-center). External devices have been added depending on the targeted setup for a particular vehicle. As knowing the position of each node is mandatory for the experiment, each setup is equipped with an USB GPS. Access to the CAN bus is provided by an USB CAN interface. Some vehicles lack CAN bus access and only transmit and/or receive the alerts generated by other vehicles. For monitoring purpose, each Airbox comes with a tablet or laptop, connected to the Raspberry Pi using its internal WiFi to forward the display.

Regarding the network, some vehicles are equipped either with an Ethernet 802.11p device (Fig. 7-top-left) or with an USB 802.11g device. Some vehicles are equipped with both and act as a gateway between both networks (this is transparent with the Airplug middleware). The infrastructure (RSU) is composed of two 802.11p gateways (Fig. 7-right) connected to a web server.

In order to orchestrate all the previously described embedded applications, the Airplug middleware is running on the Raspberries. This is a light message passing framework for intra- and inter-vehicle communication. An application may subscribe to another one to receive its messages [20,7].

6.2. Scenario

The experiment involves 10 vehicles (Fig. 7) and two road-side units. The vehicles are organized in three groups; each group has its own loop-based track to follow (Fig. 8):

- Rain Detection 1 (RD1): 3 vehicles equipped with CAN and 802.11p devices (blue track in Fig. 8);
- Rain Detection 2 (RD2): 2 vehicles equipped with CAN and 802.11p devices (orange track in Fig. 8);
- Rain Advertised (RA): 4 vehicles equipped with 802.11g devices and one equipped with both 802.11g and 802.11p devices, acting as a gateway between 802.11p and 802.11g networks (violet track in Fig. 8).

All vehicles of a group are following each other. All tracks share a round-about to enable the groups to communicate together. They also periodically get in range of an RSU. Using loop-based tracks permits to obtain a long experiment in order to confirm the results.

The heavy rain is supposed to be localized in the dashed area in Fig. 8. When approaching this area, the drivers in RD1 and RD2

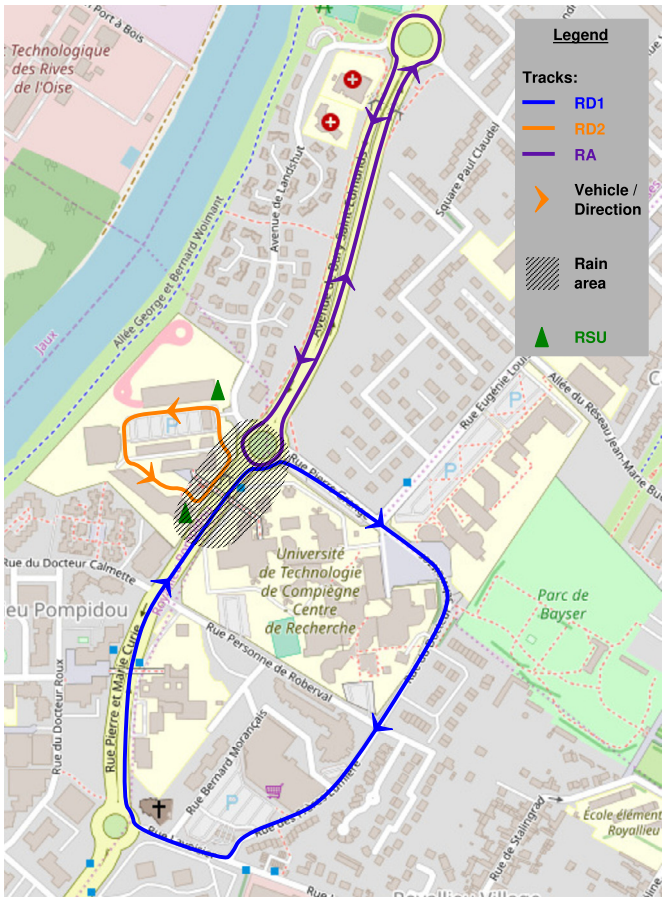

Fig. 8. Map of the proof of concept (OpenStreetMap tiles). The ten vehicles are organized in three groups, following three different tracks. Groups RD1 and RD2 will detect the rain in the dashed area. Group RA will be advertised about the rain. The RSUs forward the alert to the Internet and replay them to arriving vehicles.

turn on their wipers and progressively increase the wiper speed. The vehicles in RD1 and RD2 then cooperate to detect the hazard. The MET timer (periodic sending for the distributed data fusion) is set to 1 second; other MET parameters are given in Equations (1) and (2).

When an alert is generated by a vehicle of RD1 or RD2, it is propagated towards distant vehicles (including those of RA which are not equipped for hazard detection) and the web server. The vehicle of RA acting as a gateway receives the alert either from the RSU or a vehicle from RD1 or RD2 through the 802.11p network. It then forwards the alert to the other vehicles of RA through the 802.11g network. The complete data-flow of the system is given in Appendix A.

6.3. Results

During the experiment, all ten vehicles were working according to the scenario and they managed to detect hazards and propagate alerts to other vehicles and to the infrastructure. Even if the three groups of vehicles were not synchronized (track of different lengths, real traffic, traffic lights), the algorithms were able to work correctly, which validates the proof of concept in a real demonstration.

Alerts have been received on the web server but they have not been transmitted to web clients (case of vehicle v_7 in Fig. 2) during the experiment (this point has been tested later separately). Hence, remote vehicles were warned either by means of V2V communication or by the RSUs when they were in their communication range. This validates the role of the RSUs.

Fig. 9 shows some examples of maps available in the vehicles, provided by the graphical user interface (GUI) of the MAP application. The MAP GUI relies on OpenStreetMap tiles, which are downloaded from the Internet either in advance or when necessary (in this last case, an Internet connection is required). On these maps, the ego vehicle is displayed in red in the center. The vehicle v_3 belongs to RD1 and receives the alert from RSU_2 or v_2 before entering into the dangerous area (Fig. 9-left). The vehicle w_2 belongs to RA. It receives the alert from the vehicle w_1 and is warned about the road hazard before entering the area (Fig. 9-right).

The alerts are displayed using appropriate icons, with the related pignistic probability (Sec. 5.1). Such a probability is used to compute the saturation of the icon. In such a way, an alert with a moderate probability attracts less attention than another one with a larger probability of occurrence. In case the alert is not refreshed by the ALT application, its saturation decreases according to a decreasing function given as a parameter to MAP (actually it depends on the alert type); this ensures that the alerts eventually disappear from the map.

Besides the alerts, the other vehicles (in blue), the RSUs (green triangles) and their distance to the ego vehicle are displayed thanks to the neighbor discovery (IAL application).

As a conclusion, the complexity of the scenario and the variety of vehicles, networks and hardware used during the road tests show the feasibility of our approach. In order to study its performance, a simpler scenario is thoroughly analyzed by in-lab emulation in the next section.

7. Performance study

In this section, we present the performance study based on real-life in-lab experiments.

7.1. Setup and scenario

In-lab experiments are convenient for studying the performances of our cooperative system because they enable repeating the runs while varying parameters. For this purpose, we used the Airplug emulation tool [7], that enables high-fidelity reproduction of road experiments or creation of new in-lab experiments. The software applications running in emulation are the same as those used during a road experiment and messages are exchanged similarly. The wireless network is emulated (using the shell facilities) with some parameters that may vary, such as range, delay and reliability of the communication. Table 5 summarizes the parameters of the emulation. Fig. 10 shows an example where the proof of concept is replayed by emulation. In the following study, the local confidence is generated by MET; the CAN and CTM applications in charge of exploiting the sensor output are not used.

For the performance study, we emphasize on a convoy approaching a heavy rainfall on the N31 road near the city of Compiègne, France. The convoy is built from a real GPS track, which is duplicated eight times using an interval of 8 sec. The inter-vehicle distance is then non constant because the speed is not (e.g. it decreases in turns). A vehicle can only communicate with its neighbors if the distance is smaller than the communication range fixed to 500 m. This is a realistic multi-hop scenario for non-urban areas (which represent more than $\frac{3}{4}$ of the injuries in car accidents [43]). It is balanced because a higher vehicles density would give an advantage to both hazard detection and alert propagation while a scenario with less vehicles would lead to a partitioned network. Moreover, considering such a regular topology helps with the result analysis.

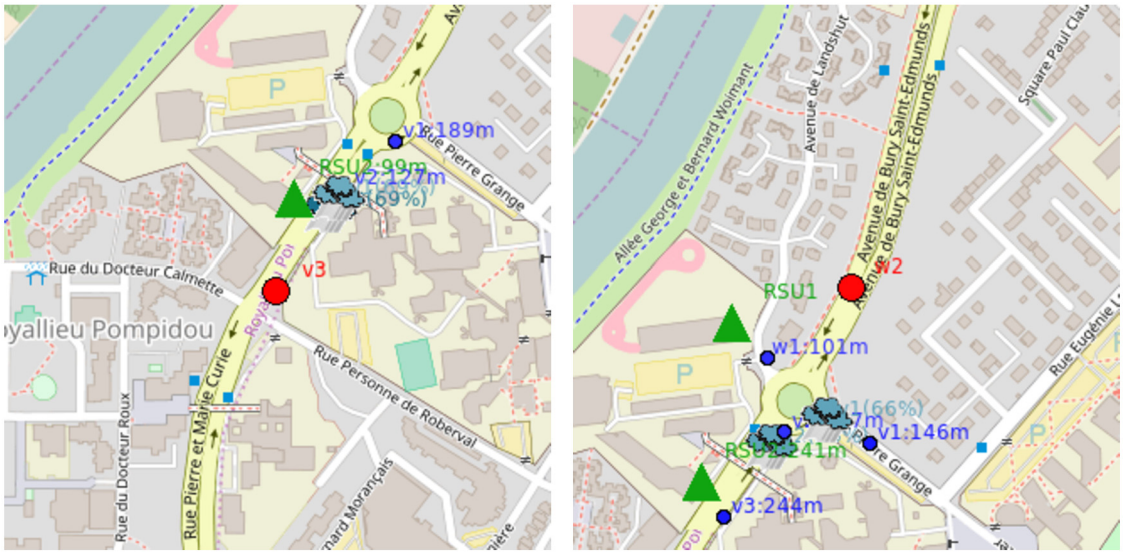

Fig. 9. Examples of maps in vehicles (v_3 on the left and w_2 on the right) provided by the GUI of the Airplug MAP application (relying on OpenStreetMap tiles). The ego vehicle is represented with a red bullet in the center. The alerts are displayed using an appropriate icon and their pignistic probability (percentage); the saturation of the icon is related to this value. When using the optional neighbor discovery, the RSU (green triangle) and the other vehicles (blue bullets) are also represented with the distance to the ego vehicle.

Fig. 11 is a screenshot of the emulation tool (relying on OpenStreetMap tiles). The vehicles are represented with blue bullets. The dashed area represents heavy rainfall. Red lines represent network connections (when they exist). The vertical bars on each node represent the local confidence (three left-most bars) and the distributed confidence (three right-most bars), expressed as pignistic probabilities. Grey is used for the *highfall* event (danger), white for the *lowfall* event (cloudy weather) and yellow for the *nofall* event (sunny weather).

The local confidence is computed using only the local input while the distributed confidence is computed thanks to the distributed data fusion. By comparing these values, the interest of the cooperative hazard detection is evaluated. For instance in Fig. 11, vehicle v_1 is under heavy rainfall, which has been detected locally and distributively (using the cooperative hazard detection) because both grey bars are large. Thought it is still far away from the hazard, vehicle v_4 becomes to be warned by the cooperative detection (*pre-alert*, see Sec. 7.3) while it is not warned by its own input: we can see that local and distributed confidences disagree on this vehicle. Additionally, a vehicle may receive an *alert* from a previous ones in the convoy, offering an anticipation period (see Sec. 7.4). Such an alert is propagated when the distributed confidence (expressed as pignistic probabilities) is above the sending threshold.

To ease reading, the following study and the plotted figures rely on the data that has been regularly saved during one run of the in-lab experiments; variations with other runs are given for comparison.

7.2. Local detection as a reference

In order to compute the anticipation period provided by our system, we need to determine the date at which the ego vehicle is able to detect the road hazard by itself. The *local detection date* denoted t_{loc} corresponds to the date of detection based on the local input of the vehicle (without any communication with others). The hazard impacts the input of the ego vehicle, then the local confidence into the feared event and in turn the related pignistic probability[14] *ploc*(*feared event*). We define the local detection date t_{loc} as the date for which *ploc*(*feared event*) > 0.5.

Fig. 12 shows the evolution of pignistic probabilities on the last vehicle (v_8) of the convoy while it is approaching the road hazard. The feared event (*highfall*) is plotted in grey (same colors as in Fig. 11). We observe the transition of the local confidence: at $t = 68$ sec, *ploc*(*nofall*) falls down and *ploc*(*lowfall*) increases. The high confidence into the feared event appears when *ploc*(*highfall*) rises sharply, defining $t_{\text{loc}}(v_8) = 76.983$ sec (green mark in Fig. 12). Similar observations have been done for the other vehicles (see the sharp transitions in Fig. 13-top). Table 6 presents the local detection dates of the eight vehicles (very similar results have been obtained in other runs with a difference up to 0.6 sec). The anticipation periods expected from the cooperative road hazard anticipation system will be computed from these dates.

7.3. Cooperative detection study

In this section, we study the performance of the cooperative hazard detection introduced in Sec. 4 independently of the cooperative alert propagation.

Fig. 13 gathers the pignistic probabilities of the *highfall* event (danger) for the eight vehicles: local confidences at the top and distributed confidences on the bottom. As all vehicles detect locally the danger roughly at the same GPS position (without any communication), Fig. 13-top gives regular plotting. By comparing with Fig. 13-bottom, we observe concomitant transitions between local and distributed confidences. This is a useful consequence of the discounting: when the ego vehicle has a large confidence into an event (using its local input), the impact on the distributed confidence computation is large because the other inputs of the computation (received values) are discounted. This property avoids false negative detection: whenever the local confidence is large, the distributed confidence is large also.

[14] For the sake of simplicity, when there is no ambiguity we refer in the rest of the analysis to *local* (or *distributed*) *confidence* instead of *pignistic probability computed from the local* (or *distributed*) *confidence*. Moreover, we note *ploc*(*event*) or *pdis*(*event*) instead of *ploc*({*event*}) and *pdis*({*event*}).

Fig. 10. Emulation of the proof of concept scenario. The computer is on the bottom-left behind the screen used to control the emulation. The bottom-right screen shows the web server receiving the alerts. The two upper screens are used to display the GUI in each vehicle (in particular the ALT and MAP applications).

The anticipation period is detected by thresholding.[15] Let T_{pre} the *pre-alert threshold*. We define the *pre-alert date* denoted t_{pre} as the date for which $pdis(feared\ event) > T_{\text{pre}}$. The pre-alert provides an anticipation period defined by $t_{\text{pre}} - t_{\text{loc}}$. This is the period before the hazard during which the ego driver can adapt its driving. A positive value is useless. In our experiment, the pre-alert threshold T_{pre} is equal to 0.25. This leads to an anticipation period for the driver around -7 ± 1.4 seconds (see Table 7), excepted for v_1 which is the first vehicle discovering the hazard (around -6.5 ± 1.8 for the ten runs). A smaller threshold would increase the anticipation period while a larger would not provide any anticipation period. However, considering low thresholds (smaller than 0.5) cannot guarantee the lack of false positives. Hence, this thresholding provides a pre-alert for the ego driver only and should not be forwarded to other vehicles.

Finally, the *distributed detection date* denoted t_{alt} is defined when the distributed confidence into the feared event becomes larger than the *alert threshold* fixed to 0.5. Though robust, the hazard detection relying on the distributed confidence ($pdis(highfall) > 0.5$) does not give any anticipation period because it happens after the local detection, as seen in Table 8 (anticipation periods are positive and then useless). Similar results are obtained for all runs.

To summarize, the cautious operator used to compute the distributed confidence avoids false positive detections by applying the *Least Commitment Principle*. Moreover, the discounting avoids false negative detections, as shown by the concomitance of the transitions between local and distributed confidences. When considering a pignistic probability larger than 0.5, the alert is confirmed but there is no anticipation. A smaller threshold T_{pre} remains useful for local pre-alerts only. Our cooperative hazard detection system is then robust and truthful but on the other hand it cannot provide any anticipation period to the driver due to is cautiousness. For this purpose, it has to be completed by an alert propagation mechanism.

7.4. Road hazard anticipation study

In this section, we study the performance of the whole road hazard anticipation system relying both on the cooperative hazard detection and the cooperative alert propagation. The combination of both ensures a reliable anticipation period for the drivers approaching the hazard.

First, it is important to notice that, by using conditional transmissions (Sec. 5) and alert-dependent conditions, the whole road hazard anticipation system remains true with respect to the alerts announced to the driver. Indeed, the upward conditions can easily be tuned so that only concerned drivers are warned about the road hazard (receiver vehicles). For instance, HOP admits conditions such as: being behind the vehicle announcing the hazard and at less than 2 km of the hazard and during 30 minutes only, etc.

Moreover, the conditional transmissions bound the area of messages propagation thanks to the forward condition. Conditions are versatile enough to address an obstacle on a given lane, an accident on a given road, a localized weather alert or a large meteorological phenomenon. This way, very few vehicles are solicited as relay and any vehicle detecting a road hazard can generate an alert without overloading the network, then enforcing the reliability of the whole cooperative system.

As explained in Sec. 5.1, an alert is propagated whenever $pdis(highfall) > T_{\text{snd}}$, where T_{snd} is the *alert sending threshold*. When dealing with alert propagation, only well confirmed alerts should be considered. Hence, the alert sending threshold satisfies $T_{\text{snd}} \geq 0.5$. The larger T_{snd} is, the more reliable the propagated alerts are but the less the anticipation period is. A too large alert sending threshold may lead a false negative. To be fair, the sending threshold T_{snd} has been set to 0.66 in our experiments. Table 9 gives the corresponding sending dates. Similar results have been obtained for all ten runs (for instance the average sending date of v_8 is 81.901 sec with a standard deviation of 0.233).

We define the *alert receiving date* $t_{\text{rcv\_}w}(v)$ as the date of reception on vehicle w of the first alert generated by vehicle v. Fig. 14 shows the alert reception dates on v_8 (red marks). For comparison, the green mark represents the local detection date on v_8 based on the local input only and used as reference date (see Table 6 and Section 7.2) and the orange mark represents the pre-alert date with $T_{\text{pre}} = 0.25$ (see Table 7 and Section 7.3). While the sending threshold has been fixed at a high value, vehicle v_8 is warned early thanks to the alert generated by v_1 ($t_{\text{rcv\_}v_8}(v_1) = 39.941$ sec) and forwarded from vehicle to vehicle in the convoy (7 hops; here, the communication range has been set to 500 m). At the same date, v_8 is not detecting any hazard with its local input (see the grey dotted line, $ploc(highfall) = 0.067$) nor with the cooperative hazard detection (see the grey solid line, $pdis(highfall) = 0.103$).

By receiving the alerts from several vehicles, the robustness of the alert propagation is reinforced while the network may be

[15] An anticipation period may be expected by analyzing the increasing of $pdis(highfall)$ before the sharp transition of the distributed confidence. However false positive may appear using this method and thresholding is preferred.

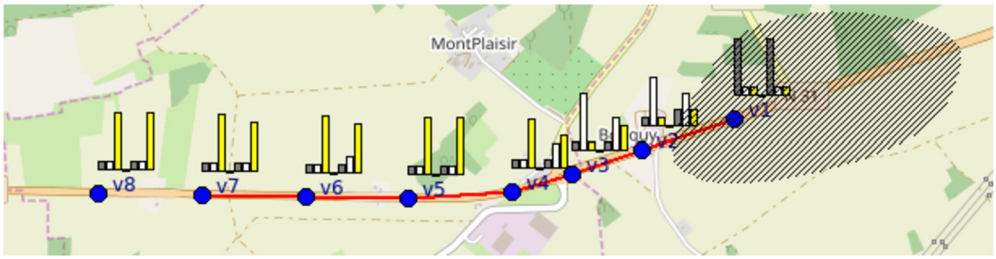

Fig. 11. Scenario of the in-lab experiment by network emulation. Vehicles move from left to right towards the rain (dashed area). On each vehicle, the three left-most bars represent the local confidence expressed as pignistic probability: grey for *highfall*, white for *lowfall* and yellow for *nofall* (sunny weather). The three right-most bars represent similarly the distributed confidence. The distributed confidence into the feared event increases while vehicles approach the rain.

Table 5
Parameters of the in-lab experiment using network emulation. The bottom part of the table is related to the analyzed run (from which the following figures come).

Parameter	Value	How
Number of vehicles	8 (convoy scenario, see Fig. 11)	Set
Number of runs	10	Set
Data fusion timer (MET)	1 sec	Set
Discounting (MET)	$x \mapsto \min(1, x + 0.1)$	Set
Upward condition (HOP)	trajectory correlation	Set
Forward condition (HOP)	trajectory correlation + distance ≤ 4 km	Set
Alert duration (ALT)	1 minute	Set
Communication range	500 m	Set
Link reliability	100%; varying from 30 to 100% in Sec. 7.5	Set
Vehicle speed	from 28 to 90 km/h, mean 81, stdev 15	Measured[a]
One hop delay	from 19.3 to 202.5 ms; mean 43.3 ms, stdev 21.6 ms	Measured[b]
Seven hop delay (convoy)	from 140 to 262 ms, mean 187 ms, stdev 39.5 ms	Measured[c]
Duration	109 sec	Measured
One hop delay	from 20 to 153 ms; mean 47.5 ms, stdev 28.4 ms	Measured[b]
Seven hop delay (convoy)	178 ms	Measured[c]
Data fusion msg sent (MET)	815	Measured
Alert messages sent (ALT)	96	Measured

[a] Measured in the recorded GPS track.
[b] Measures based on the alert messages delays at the application layer.
[c] Delay of the first alert from v_1 to v_8 at the application layer.

disconnected (inter-vehicle distance larger than the communication range) or some packets may be lost. On the other hand, the propagation has to be bound in time and space, using appropriate forward conditions with HOP. Table 10 summarizes the hazard anticipation due to the cooperative alert propagation. Results are close in the ten runs. For instance, the anticipation period on v_4 is between -12.494 and -9.329 (average of -11.759, standard deviation of 0.94). Hence, we can see that, when there are more than 4 vehicles in front of the ego vehicle (corresponding to roughly 2 km here), the cooperative alert propagation significantly increases the road hazard anticipation. This illustrates the discussion in Sec. 3.3 about the size of the area of influence of the cooperative hazard detection on the one hand and the cooperative alert propagation on the other hand (see Fig. 4): the cooperative alert propagation is useful starting from 4 hops.

As a conclusion, thanks to application-dependent conditions, the cooperative alert propagation warns only the concerned drivers. By limiting the number of relays, it allows any vehicle detecting a road hazard to propagate an alert, enforcing the reliability of the whole system despite packet loss. Moreover even when using a large sending threshold (0.66 in our experiments), the cooperative alert propagation starts being useful after few hops, providing a very significant anticipation period (reaching 37 sec in the convoy of 7 hops). With a smaller sending threshold T_{snd} (yet larger than 0.5), the anticipation period would still be larger. The local pre-alerts defined with T_{pre} appear to be useful for vehicles close to the hazard (less than 2 km in our experiment).

7.5. Impact of packet loss

When designing a cooperative application in vehicular networks, it is important to study its robustness in case of poor communication conditions. The *communication reliability* is the probability that a vehicle-to-vehicle communication succeeds. Though the convoy is sometimes disconnected in our scenario (as shown in Fig. 11 for v_7 and v_8), previous figures have been obtained with a reliability of 100%, meaning that whenever a communication involves two vehicles in their communication range, it succeeds. In this section, we study the impact of a lower network reliability, resulting in packet loss.

Fig. 15 shows the impact of the communication reliability on the distributed confidence into the feared event in vehicle v_8. The loss rate is smaller than $\frac{1}{3}$ in Fig. 15-bottom while harsh network conditions are considered in Fig. 15-top. The reference curve is here the distributed confidence $pdis(highfall)$ with the maximal reliability equal to 100% (black solid line). Contrary to the distributed confidence, the local confidence is computed without any communication, by using the local input only. Hence, $ploc(highfall)$ is not impacted by the communication reliability (black dotted line).

Unsurprisingly the more the reliability decreases, the more the distributed confidence varies (expressed as pignistic probabilities, see Note 14). Indeed the distributed algorithm is stateless: periodically, the computation is performed with the received values; in

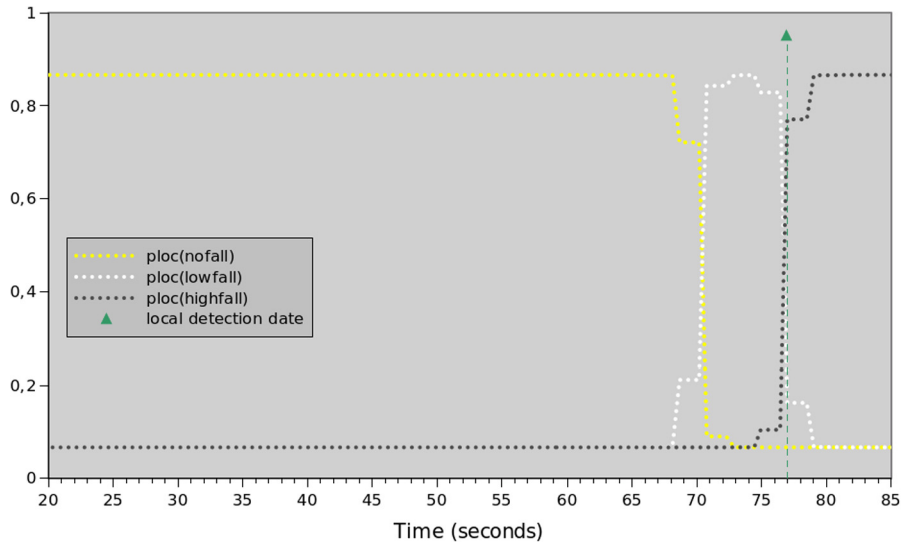

Fig. 12. Evolution of the local confidence *ploc* in vehicle v_8 approaching the road hazard. Same color code as in Fig. 11: probability of *highfall* (danger) is in grey. The local detection date (green mark) is defined when $ploc(highfall) > 0.5$.

Table 6
Local detection dates t_{loc} defined when $ploc(highfall) > 0.5$. They are used as reference for computing the anticipation period provided by the cooperative system.

Vehicle v	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
$t_{loc}(v)$ in seconds	14.444	22.995	31.994	41.803	50.609	59.581	68.267	76.983

case a message is missing the result may vary temporary.[16] Moreover, we also observe that the more the reliability decreases, the more the distributed confidence is close to the local confidence. Indeed, when taking into account disagreeing sources, the distributed confidence is smoothed compared to the local confidence. In case some messages are missing, the smoothing is less important. In case no message is received at all, only the local confidence is taken into account in the computation of the distributed confidence and both are equals.

Hence, the distributed confidences curves for a reliability smaller than 100% are mainly contained between the local confidence and the distributed confidence with rel=100%, that is between the black dotted line and the black solid line. As a consequence, the variations due to packet loss are mainly below the reference curve before the sharp transition and above the reference curve after. This leads to an interesting property: the risk of false positive pre-alert and the lack of alert sending (false negative) is neglected, even for harsh network conditions. In particular, none of the curves reaches the pre-alert threshold $T_{pre} = 0.25$ before the reference curve. Moreover none of them reach the sending threshold $T_{snd} = 0.66$ significantly after the reference curves.

Regarding the alert propagation, packet loss is usually counterbalanced by repetition. By limiting the area of distribution and the number of relay nodes, the conditional transmissions limit the bandwidth saturation and are convenient for a repetition heuristic. For this purpose, the ALT application includes several strategies such as sending alerts using burst of few messages spaced by a pause or determining the inter-packet gap using the vehicle speed. Nevertheless, such heuristics have not been used here to obtain a fair study of the packet loss impact.

As a conclusion, both hazard detection and alert propagation rely on periodic messages and stateless algorithms (newer messages withdraw the previous ones), which is advantageous to face packet loss. Moreover, by using the cautious operator, the cooperative hazard detection remains truthful against poor network conditions. By optimizing the area of distribution, the cooperative alert propagation avoids the bandwidth saturation and supports well the messages repetitions. Hence, our cooperative hazard anticipation system is robust even with poor networking conditions.

8. Discussion

In this section, the key points and the complexity of the proposed road hazard anticipation system are analyzed. Then the limitation of both the study and the system are listed. They may be improved in further work.

8.1. Key points of the system

As seen in the previous section with in-lab experiments, the robustness is an important characteristic of our system. This is due to several choices. First, the danger detection relies on data fusion, which has been designed to deal with uncertainties. It allows the dynamic of the vehicular network and the varying environmental conditions to be taken into account. Moreover, our system relies on the cautious operator, which withdraws eccentric information. Thanks to the discounting, a remote information can influence the local result only if it is confirmed by others, hence enforcing the robustness. Such cautiousness combined with a stateless distributed algorithm permits packet losses to be supported, as explained in Sec. 7.5. Finally, limiting the number of involved vehicles during the alert propagation enable alert redundancy without overloading the network, hence ensuring a robust propagation.

Another important advantage of our system is its efficiency. This is due to on-the-fly data fusion in a stateless algorithm; there is no need to collect data prior to computation and decision. Moreover, using vectors of weights simplify the computation. Finally, our alert propagation protocol is based on conditions and such

[16] Note that a stateful algorithm has the drawback of being sensitive to bad or malicious inputs that may pollute the output for a while.

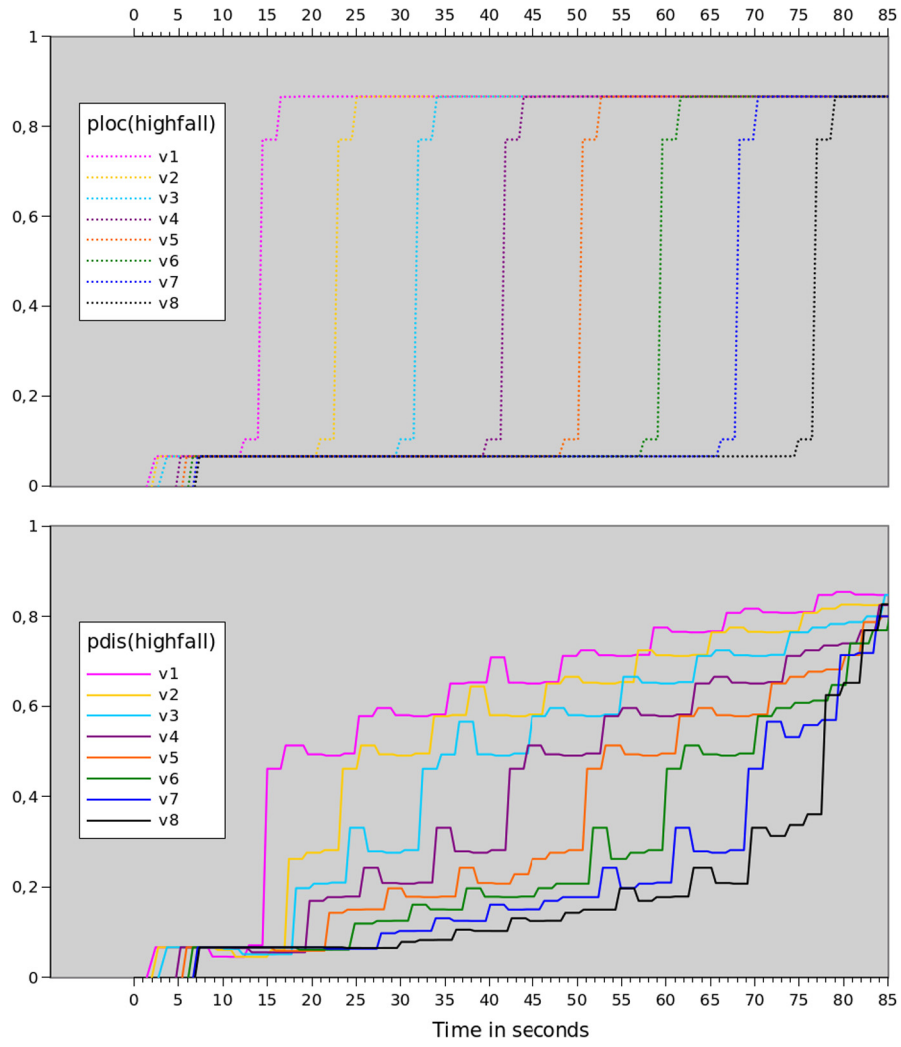

Fig. 13. Comparing the evolution of the local (top) and distributed (bottom) confidences into the *highfall* event in the convoy (expressed as pignistic probability).

Table 7
Pre-alert dates t_{pre} used to warn the ego driver only. This is the date when $pdis(highfall) > T_{pre}$. Here, the pre-alert threshold T_{pre} equals 0.25. The anticipation period is given by $t_{pre} - t_{loc}$ (positive values are then useless).

Vehicle v	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
$t_{pre}(v)$ in seconds	14.981	17.42	24.3	34.063	44.875	51.778	61.076	69.674
Anticipation period (sec)	0.537	-5.575	-7.694	-7.740	-5.734	-7.803	-7.191	-7.309

Table 8
Alert detection dates t_{alt} defined when $pdis(highfall) > 0.5$. This robust alert detection does not provide any anticipation period (positive values are useless) because $t_{alt} > t_{loc}$.

Vehicle v	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
$t_{alt}(v)$ in seconds	17.052	25.538	36.607	44.355	53.176	62.131	71.361	77.988
Anticipation period (sec)	2.608	2.543	4.613	2.552	2.567	2.550	3.094	1.005

Table 9
Alert sending dates t_{snd} defined when $pdis(highfall) > T_{snd}$. Here $T_{snd} = 0.66$.

vehicle v	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
$t_{snd}(v)$ in seconds	40.162	48.615	55.198	65.4	73.953	80.744	79.66	82.29

kind of content-addressing is more efficient than address-based propagation in a dynamic vehicular network. These properties save resources and energy.

The third advantage of our system is its architecture. It relies on a combination of cooperative strategies that complete each other: reliable danger detection combined with a selective alert propagation. The latter counterbalances the cautiousness of the former which cannot offer any anticipation alone. Moreover such algorithms neither depend on the number of vehicles nor on the state of the system. Each vehicle is independent; a driver could be warned either by its own sensors or by the cooperation of other vehicles, offering an anticipation. Also each application is

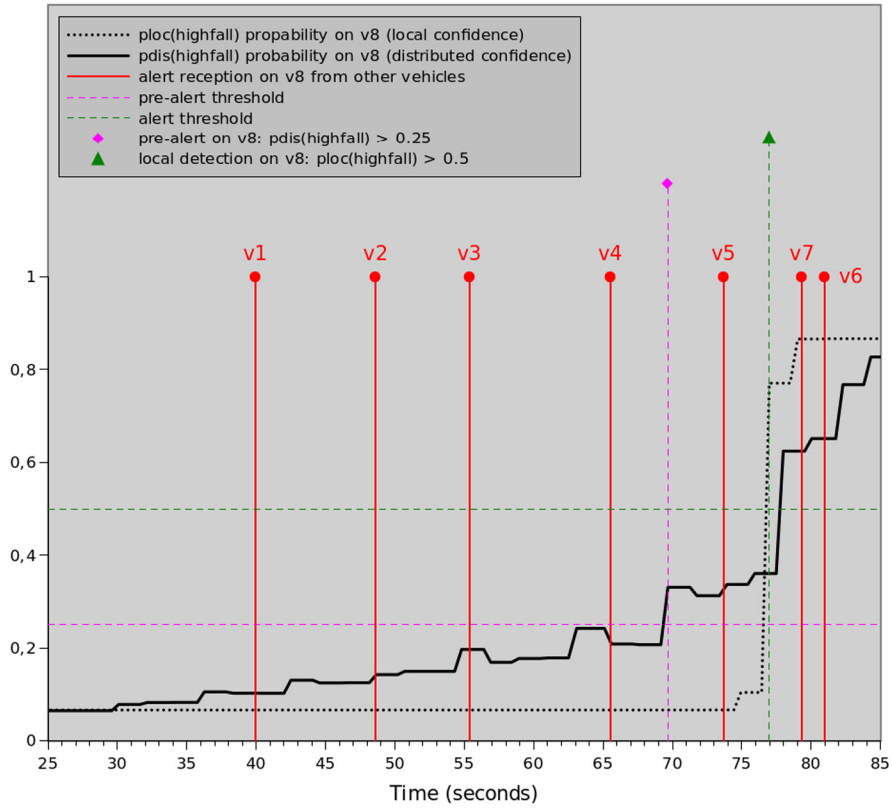

Fig. 14. Alert reception in vehicle v_8. The local confidence *ploc*(*highfall*) is plotted in dotted line and the distributed confidence *pdis*(*highfall*) in solid line. The green mark represents $t_{\text{loc}}(v_8)$, the local detection date used as reference for computing the anticipation period. The pink mark represents $t_{\text{pre}}(v_8)$, the date of pre-alert on v_8. The red marks represent $t_{\text{rcv\_}v_8}(v)$, the date of reception on v_8 of the alert generated by vehicle v.

Table 10
Alert reception dates $t_{\text{rcv\_}v_8}(v)$ on vehicle v_8 and related anticipation period defined by $t_{\text{rcv\_}v_8}(v) - t_{\text{loc}}(v_8)$ due to the cooperative alert propagation in the convoy. Here, both distributed data fusion and alert propagation are used conjointly.

Vehicle generating the alert	v_1	v_2	v_3	v_4	v_5	v_6	v_7
$t_{\text{rcv\_}v_8}(v)$ in seconds	39.941	48.622	55.355	65.52	73.687	80.988	79.342
Anticipation period (sec)	-37.042	-28.361	-21.628	-11.463	-3.296	4.005	2.359
Average of the 10 runs (sec)	-38.308	-29.785	-20.971	-11.759	-3.127	3.581	2.408

independent inside the vehicles thanks to a light message passing middleware.

8.2. Complexity analysis

The maximal convergence time of the distributed data fusion is linearly correlated to the timer of its periodic computation and the diameter of its area of influence. Such area is bounded with the discounting. Both parameters (timer and discounting function) can be tuned depending on the alert needs. In our experiment, the maximal convergence time is bounded to 10 hops $\times$ 1 second but we observed that the alert is generally detected much earlier. The convergence time of the alert propagation is related to the forwarding condition, that may include geographic and time limitation. In our experiment, it was limited to no more than 4 km from the danger and no more than 1 minute. Such parameters are adapted to local weather alerts but may be tuned for other types of alert.

The computation time is negligible for both cooperative algorithms because computations are very light. For the distributed data fusion, they consist in conversions (input, masses, weights, probabilities) and some minimum and additions on vectors of small size (8 components in our experiments). For the cooperative alert propagation, they consist in the evaluation of simple conditions.

The message complexity is related to the timer for the alert detection and to the forward condition for the alert propagation because it determines the involved vehicles. In both cases, it is parameterized and limited (1 second and vehicles up to 4 km respectively in our experiments).

Finally the memory complexity is low because our protocols are stateless. Only the received messages of the neighbors are stored during a timer. Our system is then scalable to a large number of vehicles.

8.3. Limitations of the study

A proof of concept involving 10 vehicles has been presented in Section 6. Nevertheless larger road tests could be organized, with various road traffic. As this is complex to set up, our study relies on in-lab experiments using vehicular network emulation. In future work, this performance study could be completed with more scenarios.

The in-lab study has been done with the Airplug network emulator. Thanks to this tool, the same applications (same code) are used on the road and in the lab, increasing the realism. Nevertheless the low layers are emulated and may lack accuracy. We

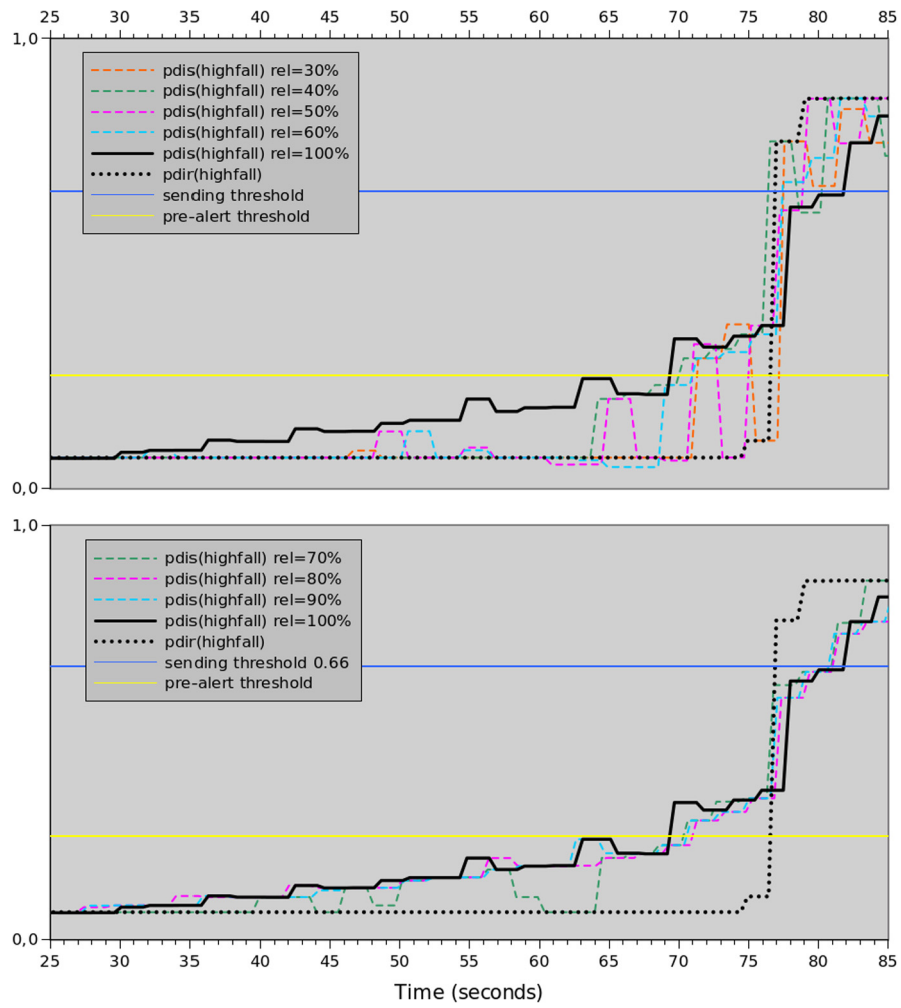

Fig. 15. Pignistic probabilities *pdis*(*highfall*) in vehicle v_8 depending on the communication reliability (low reliability on top, higher bottom). The local confidence *ploc*(*highfall*) is computed with the local inputs only (no impact of the reliability).

checked that the application delay for the alert (from the sending ALT application to the receiving ALT application) for one hop and for the whole convoy is realistic (Table 5). It is variable from a run to another (as it may also be observed on the road). We checked that such differences do not have a significant impact on the anticipation delay.

To ease the analysis, the in-lab study has been done using an artificial continuous input for computing local confidence. For instance, it may be produced by a rain detection sensor giving a continuous measurement from 0 to 1. On the other hand, to ease the road-tests, a discretized input has been used: it was built from the windscreen wipers speed because the wipers could be used even during sunny road tests. Nevertheless, we checked that, when converted into a mass function and then into a pignistic probability, the continuous input of the in-lab experiments leads to a sharp transition similar to an on-off discretized input (Fig. 12). Moreover these road and in-lab experiments show that our system can be used indifferently with continuous or discretized inputs.

8.4. Limitations of the system

Our system relies on cooperative strategies. In case a vehicle has no neighbors, it is warned by its own sensors at t_{loc} (no anticipation). To solve this issue, some RSUs could be installed in dangerous areas to upload long-term alerts to the Internet (Fig. 2); this has been tested during the road tests (Section 6). The RSUs may also relay messages between vehicles in case of sparse traffic. They may also be equipped with sensors and be involved in the cooperative strategies as a stopped vehicle would do.

Our system relies on the cautious operator to fuse the data of several vehicles. In some cases, its cautiousness may delay the alert detection. This may be solved using another idempotent data fusion operator. Another way would be to complete the system using non-idempotent operators on collected data. Our system includes the Dempster operator computation on the inputs from the neighborhood but this has not been used in this study. Note that such a functionality doubles the messages size (in our case 32 bytes more).

Our system is able to manage any kind of danger providing it is not too short nor sudden. A danger close to the ego-vehicle with a short duration (e.g. pedestrian crossing the road) may not benefit from our system. Moreover, some parameters have to be tuned for a better efficiency. In particular, the thresholds may be adapted to the event to be detected and the input provided by the related on-board sensor or calculator (local confidence). The discounting could also be adapted to the feared event to size the area of influence of the distributed data fusion (note that this impacts the convergence time). Finally the conditional transmissions may be tuned to better fit with the needs of the alert. For instance some alerts could be propagated only to arriving vehicles on the same lane (obstacle, pothole...), while some others could be propagated to a wider area (fog...). This may be done using a dictionary associating the alerts and their parameters.

9. Conclusion

This paper describes a cooperative approach to hazard anticipation in vehicular networks. It relies on a self-stabilizing distributed data fusion algorithm for the hazard detection and on a condition-based multi-hop communication protocol for the alert propagation. By using data fusion, it reinforces the confidence into the detected event. By forwarding (and uploading to the Internet) the result of a distributed cooperative hazard detection algorithm, it preserves the privacy of road users and limits the amount of collected data. By using local algorithms and on-the-fly computations, it preserves the communication and processing resources, hence minimizing energy consumption and time compared to centralized approaches.

The whole system has been demonstrated in a real-life experiment involving a large number of vehicles using cheap off-the-shelf hardware, showing the soundness and efficiency of the approach. A set of local and distributed applications has been designed on top of a light message passing middleware for this purpose. During the road tests, the participating vehicles have been warned well in advance of road hazard lying ahead.

The performance study has shown the interest of combining both the cooperative hazard detection and the cooperative alert propagation to provide significant anticipation periods before the hazard. Our in-lab experiments have confirmed the truthfulness and the robustness of the system, including the low impact of packet loss.

Future work will deal with improving our system. In particular, depending on the alert type, the cooperative hazard detection may benefit from conditional transmissions to give more importance to data fusion messages coming from vehicles ahead. This could further increase the anticipation period.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was partially supported by the Labex MS2T (ANR-11-IDEX-0004-02) and the European Celtic Plus project *CoMoSeF* (CPP2011/2-2). The authors wish to thank the CNRS, the *Direction générale de la Compétitivité, de l'Industrie et des Services* (France) and their partners, in particular Viveris Technologies who contributed to the CAN application. Particular thanks to David Bloquel for his contribution as well as all persons involved in the road tests.

This work has been carried out within the framework of the Equipex ROBOTEX (ANR-10-EQPX-44-01).

Appendix A. Data-flow

This appendix details the data-flow presented in Fig. 4. This figure can be read from left to right for the ego vehicle (from the hardware to the driver). Local Airplug messages are displayed horizontally in red while inter-vehicle Airplug messages are displayed vertically in orange.

1. Inputs
   - The embedded devices produce inputs carried over CAN frames. They are read, decoded and filtered by the CAN application, which provides a comprehensive information to CTM (in our use case, windscreen wipers speed).
   - The GNSS[5] receiver produces NMEA frames which are read by the GPS application. The latter periodically sends geographical positions and time to the subscribed application using local Airplug messages, namely IAL, ALT and MAP.
   - The GTW application detects any networking interface changes and informs HOP by means of a local Airplug message (Sec. 5.3).
2. Neighbor discovering
   - The IAL application receives local Airplug messages from GPS.
   - It exchanges inter-vehicle Airplug messages with other IAL instances in the vicinity of the ego vehicle.
   - It computes the distances to the neighbors (including the RSUs) and forwards any update to the MAP application using local Airplug messages.
3. Cooperative hazard detection (Fig. 5)
   - The CTM application formats the CAN information (Sec. 4.1) for MET as a *local confidence* (Sec. 4.2).
   - MET computes the *distributed confidence* by combining its local confidence and the distributed confidences received from its neighbors. Such distributed confidences are expressed as *vector of weights* to simplify the computation (Sec. 4.2). Periodic inter-vehicle Airplug messages are used to send the last result to the neighbors.
   - The distributed confidence is converted into *masses* for computing the *pignistic probability* and comparing it to a threshold (Sec. 5.1).
4. Cooperative alert propagation (Fig. 6)
   - ALT is warned by MET (local Airplug message) whenever the pignistic probability into the feared event is larger than a threshold.
   - It informs MAP by sending a local Airplug message.
   - MAP displays the alert on the GUI using an appropriate icon and the confidence to warn the local driver.
   - ALT also requests HOP with a local Airplug message for propagating the alert, using appropriate alert-dependent conditions.
   - HOP propagates alerts by selecting relay and destination nodes thanks to the forward and upward conditions respectively (Sec. 5.2). Inter-vehicles messages are used for this purpose; they may contain additional information required for evaluating the conditions at reception.
   - On a remote vehicle, when the upward condition is true, HOP forwards the alert to ALT using a local Airplug message.
   - ALT informs MAP by sending a local Airplug message.
   - MAP displays the alert on the GUI using an appropriate icon to warn the remote driver.

Airplug messages consist in text strings composed of key-value pairs. They have a variable size and may be completed according to options. For instance, an alert message contains a sequence number, the sender id and GPS position, an alert type and the confidence into the alert. When forwarded by HOP, it is completed with the upward and forward conditions and any data required for evaluating such conditions at reception (for instance the last GPS positions of the sender for the trajectory correlation).

Appendix B. Dempster-Shafer theory

The Dempster-Shafer theory of belief functions, also called evidence theory, has been introduced by Dempster (1968) and Shafer (1976) and has been further developed by Smets (Transferable Belief Model) in the 1980s and 1990s [11,51,56]. It is a formal framework for representing and reasoning from partial (uncertain, imprecise) information, by generalizing both the set-membership approach and the probability theory.

Information representation. Let $\Omega = \{\omega_1, ..., \omega_N\}$ be a finite set of all possible solutions to a problem. It is called the *frame of discernment*; it is composed of mutually exclusive elements. The knowl-

Table C.11
Conversion masses ↔ weights.

mass function	commonality function	weight function
$m: \mathcal{P}(\Omega) \to [0,1]$, $A \mapsto m(A)$	$q: \mathcal{P}(\Omega) \to [0,1]$, $A \mapsto q(A)$	$w: \mathcal{P}(\Omega)\setminus\Omega \to \mathbb{R}^+$, $A \mapsto w(A)$
$\sum_{A\subset\Omega} m(A) = 1$	$q(A) = \sum_{B\subset\Omega, A\subseteq B} m(B)$	$w(A) = \prod_{B\subset\Omega, A\subseteq B} q(B)^{(-1)^{\lvert B\rvert-\lvert A\rvert+1}}$

Table C.12
Examples of conversions masses ↔ weights.

	$\emptyset$	{nofall}	{lowfall}	{highfall}	{nofall, lowfall}	{nofall, highfall}	{lowfall, highfall}	Ω
m_1	0	0	0	0.8	0	0	0	0.2
q_1	1	0.2	0.2	1	0.2	0.2	0.2	0.2
w_1	1	1	1	0.2	1	1	1	-
m_2	0	0	0	0.5	0	0	0.3	0.2
q_2	1	0.2	0.5	1	0.2	0.2	0.5	0.2
w_2	1	1	1	0.5	1	1	0.4	-

edge held by one source can be quantified by a belief function defined from the power set 2^Ω to $[0,1]$. Belief functions can be expressed in several forms: the Basic Belief Assignment (BBA) denoted m, credibility function Bel, the plausibility function Pl, and the commonality function q which are in one-to-one correspondence (Appendix C).

The mass $m(A)$ quantifies the part of belief that is restricted to the proposition "the truth is $A \subseteq \Omega$ "and satisfies:

$$\sum_{A\subseteq\Omega} m(A) = 1$$

Thus, a BBA can support a set $A \subseteq \Omega$ without supporting any sub-proposition of A, which allows accounting for partial knowledge.

Another representation that can be computed from the commonality function q is the conjunctive weight function w [54,13] defined by:

$$w(A) = \prod_{B\supseteq A} q(B)^{(-1)^{|B|-|A|+1}}$$

Information fusion. Let two distinct pieces of evidence be defined over a common frame of discernment and quantified by BBAs $m_1 ... m_2$. They may be combined using a suitable operator. The most common ones are the conjunctive and disjunctive rules of combination defined, respectively, as:

$$m_{1\cap 2}(A) = \sum_{A=B\cap C} m_1(B).m_2(C)$$
$$m_{1\cup 2}(A) = \sum_{A=B\cup C} m_1(B).m_2(C)$$

The resulting BBAs should be normalized under the closed world assumption. Dempster's rule [11] denoted by $\oplus$ normalizes the result of the conjunctive rule with

$$K = \frac{1}{1 - m_{1\cap 2}(\emptyset)}$$

and sets the mass on the empty set to 0:

$$m_\oplus(A) = \begin{cases} K \cdot m_{1\cap 2}(A) & \text{for } A \neq \emptyset \\ m_\oplus(\emptyset) = 0 & \text{else.} \end{cases}$$

The conjunctive and disjunctive rules of combination are commutative and associative and assume the independence of the data sources. In [13], Denœux introduced the cautious rule of combination (denoted by $\textcircled{\scriptsize\wedge}$) to combine dependent data. This rule has the advantage of avoiding double-counting of common evidence when combining non distinct BBAs. In particular, the combination of a BBA with itself yields the same BBA: $m = m \textcircled{\scriptsize\wedge} m$ (idempotency property). The cautious rule of combination can be easily computed by taking the minimum of conjunctive weights:

$$w_{1\textcircled{\scriptsize\wedge}2} = w_1 \wedge w_2$$

where $\wedge$ denotes the minimum operator.

Reliability and discounting factor. The belief function framework makes it possible to model the user's opinion about the reliability of a source [56]. The idea is to weight more heavily the opinions of the best source and conversely for the less reliable ones. The result is a discounting of the BBA m produced by the source, resulting in a new BBA m^α defined by:

$$m^\alpha(A) = \begin{cases} \alpha \cdot m(A) & \forall A \subsetneq \Omega \\ 1 - \alpha + \alpha \cdot m(\Omega) & \text{if } A = \Omega \end{cases}$$

where $\alpha \in [0,1]$. This discounting factor can be regarded as the degree of trust assigned to the sensor.

Decision-making. There are several strategies for decision making with belief functions [14]. An optimistic strategy consists in selecting the maximum of plausibility function to singletons. The advantage is the low computation complexity. A strategy widely used is to select as solution the singleton ω with the maximum of pignistic probability transformation defined by:

$$BetP(\omega) = \sum_{A\subseteq\Omega, \omega\subseteq A} \frac{m(A)}{|A|} \qquad \forall \omega \in \Omega$$

which is equivalent to Eq. (3). Using this strategy allows comparing results with probabilistic models.

Appendix C. Conversion masses-weights

To optimize the on-the-fly computation of the distributed confidence with the cautious operator, our algorithm relies on weights. This is another representation of belief functions, which are obtained from masses using commonalities [54,12], as summarized in Table C.11.

Applying such formulae to the masses function m_1 and m_2 introduced in Section 4.2 as example, we obtain the results given in Table C.12. The computation is implemented with about 40 lines of Tcl code to obtain commonalities from masses and 60 to obtain weights from commonalities. The reverse computation is done in about 185 lines.

References

[1] 3GPP message analyzer, http://www.3gpp-message-analyser.com/decoder/msd.htm, 2021, Consult on November.

[2] Sassan Ahmadi, Chapter 7 - vehicle-to-everything (V2X) communications, in: Sassan Ahmadi (Ed.), 5G NR, Academic Press, 2019, pp. 789–843.

[3] Saeid Pourroostaei Ardakani, Chiew Foong Kwong, Pushpendu Kar, Qianyu Liu, Lincan Li, Cnn: a cluster-based named data routing for vehicular networks, IEEE Access 9 (2021) 159036–159047.

[4] A. Bachir, A. Benslimane, A multicast protocol in ad hoc networks inter-vehicle geocast, in: The 57th Semiannual Vehicular Technology Conference (VTC 2003-Spring), vol. 4, April 2003, pp. 2456–2460.

[5] A. Bonyár, A. Géczy, O. Krammer, H. Sántha, B. Illés, J. Kámán, Z. Szalay, P. Hanák, G. Harsányi, A review on current ecall systems for autonomous car accident detection, in: 2017 40th International Spring Seminar on Electronics Technology (ISSE), 2017, pp. 1–8.

[6] L. Briesemeister, L. Schafers, G. Hommel, Disseminating messages among highly mobile hosts based on inter-vehicle communication, in: Proceedings of the IEEE Symposium on Intelligent Vehicles (IV 2000), 2000, pp. 522–527.

[7] A. Buisset, B. Ducourthial, F. El Ali, S. Khalfallah, Vehicular networks emulation, in: 19th International Conference on Computer Communication Networks (ICCCN), August 2010.

[8] C. Cooper, D. Franklin, M. Ros, F. Safaei, M. Abolhasan, A comparative survey of VANET clustering techniques, IEEE Commun. Surv. Tutor. 19 (1) (2017) 657–681.

[9] F. Cunha, L. Villas, A. Boukerche, G. Maia, A. Viana, R.A.F. Mini, A.A.F. Loureiro, Data communication in VANETs: survey, applications and challenges, Ad Hoc Netw. 44(C) (December 2016) 90–103.

[10] A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping, Ann. Math. Stat. 38 (1967) 325–339.

[11] A.P. Dempster, A generalization of Bayesian inference, J. R. Stat. Soc. 30 (1968) 205–247.

[12] T. Denœux, Conjunctive and disjunctive combination of belief functions induced by non distinct bodies of evidence, Artif. Intell. 172 (2008) 234–264.

[13] T. Denœux, Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence, Artif. Intell. 172 (2008) 234–264.

[14] Thierry Denœux, Decision-making with belief functions: a review, Int. J. Approx. Reason. 109 (2019) 87–110.

[15] A.F.F. Joao Dias, J.P.C. Joel Rodrigues, Neeraj Kumar, Kashif Saleem, Cooperation strategies for vehicular delay-tolerant networks, IEEE Commun. Mag. 53 (12) (2015) 88–94.

[16] Amit Dua, Neeraj Kumar, Seema Bawa, A systematic review on routing protocols for vehicular ad hoc networks, Veh. Commun. 1 (1) (2014) 33–52.

[17] B. Ducourthial, V. Cherfaoui, Experiments with self-stabilizing distributed data fusion, in: 35th IEEE Symposium on Reliable Distributed Systems, SRDS 2016, Budapest, Hungary, September 26–29, 2016, 2016, pp. 289–296.

[18] B. Ducourthial, V. Cherfaoui, T. Denœux, Self-stabilizing distributed data fusion, in: Stabilization, Safety, and Security of Distributed Systems, in: Lecture Notes in Computer Science, vol. 7596, Springer, Berlin Heidelberg, 2012, pp. 148–162.

[19] B. Ducourthial, Y. Khaled, M. Shawky, Conditional transmissions: performance study of a new communication strategy in VANETs, IEEE Trans. Veh. Technol. 56 (6) (Nov 2007) 3348–3357.

[20] B. Ducourthial, S. Khalfallah, A platform for road experiments, in: VTC Spring 2009 - IEEE 69th Vehicular Technology Conference, 2009, pp. 1–5.

[21] N. El Zoghby, V. Cherfaoui, B. Ducourthial, T. Denœux, Distributed data fusion for detecting sybil attacks in VANETs, in: Springer-Verlag (Ed.), Proceedings of the 2nd International Conference on Belief Functions, Advances in Intelligent and Software Computing, 2012.

[22] Yaser P. Fallah, ChingLing Huang, Raja Sengupta, Hariharan Krishnan, Design of cooperative vehicle safety systems based on tight coupling of communication, computing and physical vehicle dynamics, in: Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 159–167.

[23] Bruno Fernandes, Muhammad Alam, Vitor Gomes, Joaquim Ferreira, Arnaldo Oliveira, Automatic accident detection with multi-modal alert system implementation for its, Veh. Commun. 3 (2016) 1–11.

[24] Marco Galvani, History and future of driver assistance, IEEE Instrum. Meas. Mag. 22 (1) (2019) 11–16.

[25] David Hall, Chee-Yee Chong, James Llinas, Martin Liggins, Distributed Data Fusion for Network-Centric Operations, 1st edition, CRC Press, Inc., USA, 2012.

[26] Ruiyan Han, Jinglun Shi, Quansheng Guan, Farhad Banoori, Weiqiang Shen, Speed and position aware dynamic routing for emergency message dissemination in vanets, IEEE Access 10 (2022) 1376–1385.

[27] Elwan Héry, Philippe Xu, Philippe Bonnifait, Consistent decentralized cooperative localization for autonomous vehicles using lidar, gnss, and HD maps, J. Field Robot. 38 (4) (2021) 552–571.

[28] Ling Chau Hua, Mohammad Hossein Anis, Por Lip Yee, Muhammad Alam, Social networking-based cooperation mechanisms in vehicular ad-hoc network—a survey, Veh. Commun. 10 (2017) 57–73.

[29] Tiffany Hyun-Jin Kim, Ahren Studer, Rituik Dubey, Xin Zhang, Adrian Perrig, Fan Bai, Bhargav Bellur, Aravind Iyer, VANET alert endorsement using multi-source filters, in: ACM VANET, Chicago, Illinois, USA, September 2010.

[30] Intelligent Transport Systems (ITS), Vehicular communication; basic set of applications; part 3: Specifications of decentralized environmental notification basic service, Technical Report EN 302 637-3, ETSI, 2013.

[31] Intelligent Transport Systems (ITS), Vehicular communication; basic set of applications; part 2: Specification of cooperative awareness basic service, Technical Report EN 302 637-2, ETSI, November 2014.

[32] Intelligent Transport Systems (ITS), Vehicular communications; geonetworking; part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; sub-part 1: Media-independent functionality, Technical Report EN 302 636-4-1, ETSI, August 2017.

[33] Hwanseok Jeong, Yiwen Shen, Jaehoon Jeong, Tae Oh, A comprehensive survey on vehicular networking for safe and efficient driving in smart transportation: a focus on systems, protocols, and applications, Veh. Commun. 31 (2021) 100349.

[34] K. Kamakura, B. Ducourthial, Experimental validation of cooperative approach near road side units, in: International Wireless Communications and Mobile Computing Conference, IWCMC 2014, Nicosia, Cyprus, August 4–8, 2014, 2014, pp. 1010–1015.

[35] H. Li, F. Nashashibi, M. Yang, Split covariance intersection filter: theory and its application to vehicle localization, IEEE Trans. Intell. Transp. Syst. 14 (4) (Dec 2013) 1860–1871.

[36] H. Li, G. Zhao, L. Qin, H. Aizeke, X. Zhao, Y. Yang, A survey of safety warnings under connected vehicle environments, IEEE Trans. Intell. Transp. Syst. (2020).

[37] Antoine Lima, Philippe Bonnifait, Véronique Cherfaoui, Joelle Al Hage, Data fusion with split covariance intersection for cooperative perception, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2021, pp. 1112–1118.

[38] S. McLaughlin, S. Krishnamurthy, V. Challa, Managing data incest in a distributed sensor network, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003), Hong Kong, 2003, p. V–269.

[39] F. Mira Bou, D. Mercier, F. Delmotte, E. Lefèvre, Methods using belief functions to manage imperfect information concerning events on the road in VANETs, Transp. Res., Part C, Emerg. Technol. 67 (2016) 299–320.

[40] H.B. Mitchell, Multi-Sensor Data Fusion, SPRINGER, 2007.

[41] Directorate-General Mobility and Unit C2 – Road Safety Transport. Road safety in the European Union – trends, statistics and main challenges, Technical report, European Commission April 2018.

[42] European Road Safety Observatory, Annual accident report, Technical report, European Commission, 2018.

[43] European Road Safety Observatory, Annual statistical report on road safety in the EU, Technical report, European Commission, 2020.

[44] European Road Safety Observatory, Road safety targets, monitoring report 2000-2021, Technical report, European Commission, May 2021.

[45] Sarah Oubabas, Rachida Aoudjit, Joel J.P.C. Rodrigues, Said Talbi, Secure and stable vehicular ad hoc network clustering algorithm based on hybrid mobility similarities and trust management scheme, Veh. Commun. 13 (2018) 128–138.

[46] H. Pimenta de Moraes Jr, B. Ducourthial, Cooperative neighborhood map in VANETs, in: 33rd ACM/SIGAPP Symposium on Applied Computing (SAC 2018), Pau, France, April 2018, pp. 2109–2116.

[47] J. Radak, B. Ducourthial, V. Cherfaoui, S. Bonnet, Detecting road events using distributed data fusion: experimental evaluation for the icy roads case, IEEE Trans. Intell. Transp. Syst. 17 (1) (2016) 184–194.

[48] M. Raya, P. Papadimitratos, V.D. Gligor, J.-P. Hubaux, On data-centric trust establishment in ephemeral ad hoc networks, in: IEEE INFOCOM 2008 - the 27th Conference on Computer Communications, 2008, pp. 1238–1246.

[49] Paulo H.L. Rettore, Bruno P. Santos, Roberto Rigolin F. Lopes, Guilherme Maia, Leandro A. Villas, Antonio A.F. Loureiro, Road data enrichment framework based on heterogeneous data fusion for its, IEEE Trans. Intell. Transp. Syst. 21 (4) (2020) 1751–1766.

[50] Johan Scholliers, Mikko Tarkiainen, Anne Silla, Michiel Modijefsky, Rick Janse, Guus van den Born, Study on the feasibility, costs and benefits of retrofitting advanced driver assistance to improve road safety, executive summary, Technical report, Directorate-General for Mobility and Transport, Directorate C — Land Unit C.2 — Road Safety, European Commission, February 2020.

[51] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.

[52] Hamayoun Shahwani, Syed Attique Shah, Muhammad Ashraf, Muhammad Akram, Jaehoon (Paul) Jeong, Jitae Shin, A comprehensive survey on data dissemination in vehicular ad hoc networks, Veh. Commun. 34 (2022) 100420.

[53] Pranav Kumar Singh, Sunit Kumar Nandi, Sukumar Nandi, A tutorial survey on vehicular communication state of the art, and future research directions, Veh. Commun. 18 (2019) 100164.

[54] Ph. Smets, The canonical decomposition of a weighted belief, in: Int. Joint Conf. on Artificial Intelligence, Morgan, San Mateo, Ca, 1995, pp. 1896–1901, Kaufman.
[55] Ph. Smets, Decision making in the TBM: the necessity of the pignistic transformation, Int. J. Approx. Reason. 38 (2005) 133–147.
[56] Philippe Smets, Robert Kennes, The transferable belief model, Artif. Intell. 66 (1994) 191–234.
[57] Patrick Sondi, Lucas Rivoirard, Martine Wahl, Supporting multiple cooperative applications through vehicle-to-vehicle communications, in: 2019 Global Information Infrastructure and Networking Symposium (GIIS), 2019, pp. 1–6.
[58] Technical Committee Intelligent Transport Systems. Vulnerable road users (VRU) awareness, Technical Report TS 103-300, ETSI, May 2020.
[59] F. Terroso-Sáenz, M. Valdés-Vela, C. Sotomayor-Martínez, R. Toledo-Moreo, A.F. Gómez-Skarmeta, A cooperative approach to traffic congestion detection with complex event, IEEE Trans. Intell. Transp. Syst. 13 (2) (2012).
[60] S.S. Thomas, S. Gupta, V.K. Subramanian, Event detection on roads using perceptual video summarization, IEEE Trans. Intell. Transp. Syst. 19 (9) (Sep. 2018) 2944–2954.
[61] Rojin Tizvar, Maghsoud Abbaspour, A density-aware probabilistic interest forwarding method for content-centric vehicular networks, Veh. Commun. 23 (2020) 100216.
[62] N. Ullah, X. Kong, Z. Ning, A. Tolba, M. Alrashoud, F. Xia, Emergency warning messages dissemination in vehicular social networks: a trust based scheme, Veh. Commun. 22 (2020) 100199.
[63] Ting Yin, Decai Zou, Xiaochun Lu, Cheng Bi, A multisensor fusion-based cooperative localization scheme in vehicle networks, Electronics 11 (4) (2022).
[64] Y. Zeng, D. Li, A.V. Vasilakos, Opportunistic fleets for road event detection in vehicular sensor networks, Wirel. Netw. 22 (2016) 503–521.
[65] Hui Zhang, Xinming Zhang, Dan Keun Sung, An efficient cooperative transmission based opportunistic broadcast scheme in vanets, IEEE Trans. Mob. Comput. (2021) 1–1.