Early Detection of Dangerous Events on the Road Using Distributed Data Fusion

Jovan Radak, Bertrand Ducourthial and Véronique Cherfaoui
Heudiasyc Laboratory, Université de Technologie de Compiègne, Compiègne, France
Email:jovan.radak@hds.utc.fr, bertrand.ducourthial@hds.utc.fr, veronique.cherfaoui@hds.utc.fr

***Abstract*—Intelligent transport systems are a fast developing area of research with great impact on everyday life. One of the main ideas in this area is to use all possible information, coming from vehicles and the infrastructure, in order to make the system "smarter" and avoid unwanted situations – collisions, accidents, bottlenecks... Sources of data are sometimes unreliable and may lead vehicles or the whole system to wrong conclusions and adaptations. We are presenting the application of a distributed data fusion algorithm to detect dangerous events on the road. This application is adapted to detect the possibility of encountering icy roads based on readings from wireless temperature sensors. It takes into account data not only directly obtained from sensors but also from the neighborhood of each element in the system. This way, we obtain a more robust solution that is flexible to the unreliabilities of the sources of data. We demonstrate the possibility to deduce proper results from unreliable data. The algorithm is tested in emulation, using data from a real testbed, to show the usefulness and correctness of our approach.**

Keywords: Vehicular networks, distributed data fusion, road-side units, wireless sensors, testbed, emulation, experiments.

I. Introduction

A. Motivation

Enabling vehicles and additional communication infrastructure with "smart" algorithms may help solving some of the major safety problems [6] in vehicular networks. In early works in this area researchers noted the importance of smart cars and the challenges to build them [17]. In general, the intelligence of vehicles heavily depends on a large amount of gathered data that is processed and then used to give drivers or vehicles useful information. This data may come from different sources and by its nature is heterogeneous. To process this large amount of data some form of data fusion has to be applied. Many different data fusion techniques are proposed [10], [11], depending on the specific problem that they solve and the sources of data [9]. Fusion of information can be used in different applications, from various positioning problems [8] to detecting false nodes in networks [7]. This approach is specially suitable for future smart cities equipped with a large number of sensors that may act as data sources not only for their citizens but for vehicles as well.

One of the main problems in the fusion of large amount of data is the inaccuracy and reliability of data. It is relatively easy to come up with a conclusion how to mix two pieces of information when the data sources are homogeneous, reliable and give similar readings. However, in the case of conflicting or missing data sources it may be problematic to reach a decision. The theory of belief functions have this property which makes it suitable in the presence of imprecise, uncertain and incomplete data [4].

In this paper, we present a distributed data fusion that is based on the theory of belief functions [5] and its generic algorithm. We designed an application dedicated to the early detection of dangerous events on the road. This application relies on distributed data fusion taking into account conflicting pieces of information and applies calculations until it reaches a decision. We show that the application correctly assesses the risk of dangerous icy roads even when some sensors give wrong or conflicting information. Hence, this technique surpasses a simple alert diffusion and is more robust to deficient sensors.

B. Related work

Presence of numerous sources of data are making decision problems even more complex. The problem of information fusion is well studied in various research areas (sensor networks, image processing, etc). Different solutions were proposed to deal with large amount of data sources in various applications [11]. Another important aspect is impact of driver's behavior on the functioning of whole system [3]. In [12] authors demonstrated the importance of early and appropriate signaling to drivers of possibly dangerous events on the road.

Interesting solution that brings in the connection of in-car mobile applications, drivers and hazardous events is presented in [13]. Authors have developed a system that detects speed bumps and potholes based on synchronized sensor readings and extraction from a video feed that is taken simultaneously with sensor readings. Although an interesting solution, this is limited to the usage of smartphones and specific capabilities of certain types of mobile phones. This study is also limited in terms of data dissemination, *i.e.* it is not clear how other users can make use of this data and if some calculations can be done cooperatively. In [15] authors present a testbed that enables communication between vehicles and infrastructure. This testing infrastructure gives interesting results and shows the significance of this approach. However, due to the different scope of the article, authors do not cover the possibility of equipping the infrastructure with sensors that may communicate with vehicles.

We are presenting a solution that links these two interesting points. We investigate the possibility of using multiple sources of the same data type to deal with uncertainties. We show how, with the help of distributed data-fusion algorithms, we can make use of this data. Finally, a solution that gives an assessment of the risk of ice on the road is proposed. We discuss results and how this approach can be used to give drivers early warnings on the possibly dangerous events on the road.

We start our presentation defining the basic concepts of Belief functions and the principle of distributed data fusion. Then, we detail our application of the distributed data fusion algorithm for dangerous event detection (Section II). In Section III we explain how this application has been implemented as well as the whole detection system relying on sensors, road-side units, vehicles and dedicated software. Section IV presents our experimental study. Extensive study has been done thanks to a network emulator that reproduces the whole testbed as well as with real data obtained from the testbed.

II. Distributed data fusion for dangerous event detection

Data fusion can be described as a mechanism that combines data retrieved from different sources and further on reduces uncertainty on the gathered data or generates decisions based on the obtained data. In general, data fusion is either centralized (collecting data and applying an algorithm in some kind of central unit) or distributed (each unit is capable of applying the algorithm on data gathered locally – from the neighborhood). In the remainder of this section we present the basic notions of the Belief functions theory, a generic distributed data fusion algorithm based on this framework and the application of distributed data fusion to detect dangerous events on the road.

A. Theory of belief functions

The theory of belief functions or Dempster-Shafer theory is one of the widely-used frameworks in data fusion [9]. This theory is specially suitable for modeling uncertainties and the lack of information [16]. This theory generalizes the probability theory and the possibility theory. In the Dempster-Shafer theory, a set $\Omega = \omega_1, ..., \omega_n$ of mutually exclusive propositions is called the frame of discernment. The main difference, compared to the probability theory is the fact that the mass of evidence is attributed not only to single hypotheses ω_i, but to any subset of Ω, including the empty set.

Following the general framework for the belief representation [4], we define the state of belief of node v on the global frame of discernment Ω. A state of belief is assigned using *basic belief assignment* that is most commonly represented as a *mass function*, m^Ω. A mass function m^Ω is a mapping from 2^Ω (the set of subsets of Ω) to the interval $[0, 1] \in \mathbb{R}$. The sum of all masses is equal to 1. A mass value is directly proportional to confidence, the more the node is confident in $A \subset \Omega$, the higher is $m^\Omega(A)$. The masses can be combined with different types of operators such as the *conjunctive operator* which emphasizes agreement when it combines two mass functions that are reliable and independent or the *Dempster's operator* that ignores conflict, spreading it to the other sets. These two operators are associative and commutative. In this work, we are using the *cautious operator* [2]. This operator is associative, commutative and idempotent. Idempotency is an important property because it allows data coming from sources that are not independent to be combined.

In the following, direct confidence refers to mass values of the data obtained solely from the node itself, while distributed confidence refers to the combination of the mass values from the node itself with those received from its neighbors. It thus combines all data sources in the network.

B. Distributed data fusion

We can now summarize the principle of the generic data fusion algorithm introduced in [5]. This algorithm supposes that each node computes a mass function called its *direct confidence* starting from a local measurement from an external source of data (sensor). The *basic belief assignment* is done using sigmoid-like functions. These functions are used to map the measurement to a mass function, spreading confidence on focal elements, subsets of the frame of discernment Ω. The direct confidence (private mass values) is updated whenever a new measurement is produced by the local sensor.

Each node computes its *distributed confidence* periodically by combining its own direct confidence with the output of its neighbors. In order to solve the data incest problem, the cautious operator is used. Data incest appears when a single piece information from the same source is used in the fusion process more than once. Discounting is applied to each received distributed confidence to give the closer source a higher priority than the farther one. Thus, a distant source of information will still be taken into account to compute the distributed confidence of a given node but with less importance because its output will be discounted at each hop. As a consequence, the distributed confidence computed on a node is different from another node because while it takes into account the same sources of information (direct confidences of each node of the network), they are discounted differently depending on the position of the node in the network. The result reflects the local situation in the vicinity of the node without ignoring information from other, more distant, nodes. The idempotency of the cautious operator and the discounting operations ensure the convergence of the algorithm.

Then the distributed confidence is broadcast in the neighborhood. Each node stores the distributed confidences received from its neighbors until the next local computation. Hence, the direct confidence of a node will be taken into account by its neighbors, then by the neighbors of its neighbors and so on. In this way each node is contributing to the computation of the distributed confidence.

To summarize the behavior of this distributed algorithm, every node periodically computes its direct confidence from a local measurement using sigmoid-like functions. It also stores the distributed confidences sent by its neighbors. It periodically computes its own distributed confidence using its direct confidence and the last received distributed confidences from its neighbors. Then it broadcasts its direct confidence in the neighborhood. Old received messages are deleted after a delay of several periods of computation. Keeping old messages allows the output of a neighbor to be taken into account even if some message losses occur due to problems in communication. Nodes are not synchronized, they rather use their own timers.

C. Robust application for detection of icy road

In this work we focus on the application of distributed data fusion to detect icy roads. Our aim is to have information that can be used to warn the drivers when they approach a slippery road. This application allows us to illustrate the usefulness of distributed data fusion for intelligent transportation systems. Similar applications could be designed for other road hazards. While it is clear that the road is dangerous for some temperatures

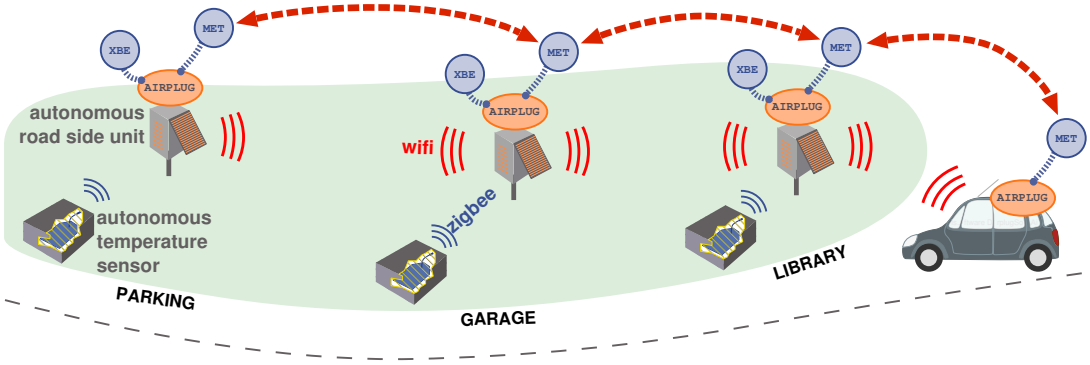


Fig. 1: Testbed with xbee sensors, road-side units and software support from Airplug platform

and not for others, there exists a range of temperatures where there is a doubt. In here we define three states for the road, as a first approach, corresponding roughly to three ranges of temperatures. These states lead to the definition of the following frame of discernment: $\Omega = \{freeze, slip, safe\}$.

We have assigned basic belief assignments from temperature sensor measurements using sigmoid functions defined for the subsets of the Ω set. The idea is to give a large mass to the state corresponding to the measured temperature and less to others. The closer the measurement gets to a threshold between two states, the more the mass is spread over several subsets of Ω. Moreover, we do not completely trust the sensors and a mass is always given to the subset Ω itself, representing the doubt (given as α). When affecting the masses for the direct confidence, there is no values for the subset $\emptyset$ because it represents the conflict. However some values may appear for $\emptyset$ after combination of several sources in the distributed confidence. The thresholds and referent values that define the road states are chosen in such a way that they correspond to real values. These parameters can be changed easily so that we can create a model that can be tested with higher temperatures (than those that produce frozen road). Sigmoid functions are given as:

$$m\{freeze\} = (1-\alpha) - \frac{1-\alpha}{1-e^{-\lambda(T_{cur}-T_{ref}+T_{thr1})}} \quad (1)$$

$$m\{slip\} = \frac{1-\alpha}{1-e^{-\lambda(T_{cur}-T_{ref}+T_{thr1})}} - \frac{1-\alpha}{1-e^{-\lambda(T_{cur}-T_{ref}-T_{thr1})}} \quad (2)$$

$$m\{slip, safe\} = \frac{1-\alpha}{1-e^{-\lambda(T_{cur}-T_{ref}-T_{thr1})}} - \frac{1-\alpha}{1-e^{-\lambda(T_{cur}-T_{ref}-T_{thr2})}} \quad (3)$$

$$m\{safe\} = \frac{1-\alpha}{1-e^{-\lambda(T_{cur}-T_{ref}-T_{thr2})}} \quad (4)$$

$$m\{\Omega\} = \alpha \quad (5)$$

where T_{cur} is current temperature – value read from the sensor, T_{ref}, T_{thr1}, T_{thr2} and λ are the reference values chosen in such a way that sigmoids correspond to real values and have appropriate shapes. The value $\alpha = 0.2$ corresponds to the belief that none of the values for the temperature is correct (doubt). We have used parameters for sigmoid functions in such a way that they reflect actual possibilities for the frozen road condition. For our purposes, the road is $safe$ for temperatures above $7°C$, road status is between $safe$ and $slip$ in the interval $\{3°C, 7°C\}$, road is most probably in the $slip$ state in the temperature interval $\{-1°C, 3°C\}$ and most probably in the $freeze$ state for temperatures below $-1°C$. The sigmoid functions shapes for these referent values can be seen in the Section IV on Figure 7(b) where local values are decreasing linearly with time allowing direct confidence to take the shape of sigmoid functions.

III. Implementation of the detection system

In the previous section, we introduced our distributed data fusion application for frozen road detection. In this section, we present the implementation of the whole detection system relying on sensors, road-side units, vehicles and related software. We first present the overall architecture. Then we describe the hardware components of the system. Next we introduce the framework used to implement the applications before describing the software components of the system.

A. System architecture

Our detection system relies on stationary and mobile temperature measurements combined using our distributed data fusion application described in the previous section. The fix measurements are performed by wireless sensors close to the road that send their measurements to some solar road-side units (Figure 2). The road side units communicate between themselves and with equipped vehicles in their transmission range. Vehicles used in the testbed are equipped with hardware units allowing them to communicate with road-side units. They broadcast their own confidence about the road state, calculated from their own temperature and those received from the RSU.

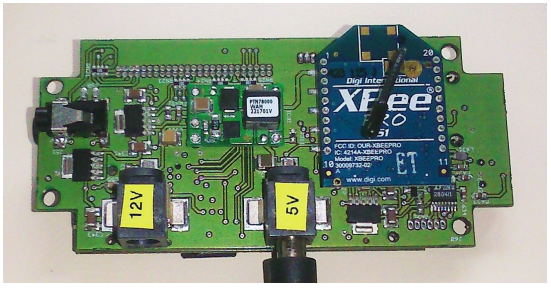
The RSUs use wifi to communicate between themselves and with vehicles. Messages that are exchanged in this communication are those of the distributed data fusion application, namely the distributed confidence that is computed periodically by each node. Communication with sensors relies on dedicated packets exchanged with zigbee modules – that both wireless sensors and RSUs have.

B. Hardware components

For development and testing purposes we have designed and deployed the hardware of the testbed that consists of cheap off-the-shelf wireless sensors and dedicated road-side units, called Airbox (developed in our laboratory).

The testbed is deployed using 3 road-side units and 3 Xbee sensors communicating with them wirelessly via the 802.15.4 protocol. Data is gathered using the serial ports of the RSUs. We have developed a support program that is used to connect computing units (RSUs or regular PCs) with wireless sensors

(a) Xbee sensor (with xbee modem)



(b) Airbox – IGEP based development boards, used as RSU

(c) RSU enclosed with energy harvesting unit and Xbee sensor

Fig. 2: Hardware elements of the detection system

and gather data through the Xbee modem connected to the serial port of the RSU [14].

Wireless sensors[1] are equipped with a temperature sensor. We have deployed a simple topology – using exactly one sensor for each RSU. The support program allows more complicated scenarios with multiple sensors per RSU but this strictly depends on the application for which this testbed is used.

The Airbox units are based on the IGEP platform[2] – a TI OMAP controller with 512 MB of RAM. They are running the Linux operating system and a software suite developed for these purposes. The driver part of the support program is responsible for establishing a connection between the Airbox serial port, the Xbee module and the Xbee sensor. The application exchanges binary packets between the RSU and the wireless sensors and decodes them upon reception. Decoded data is further transfered to other applications that may use them.

C. Airplug framework

The Airplug software distribution is a program suite based on the Airplug framework aiming to fully support simulation, development, testing and deployment of dynamic networks. This is a simple framework based on a few development rules that focus on the implementation of portable software for highly dynamic networks. These rules are: *(i)* usage of standard input/output system to ensure independence of programming language implementation, *(ii)* use of standard ASCII text messages to ensure portability (possibility of usage in different operating systems); *(iii)* simple message addressing scheme which includes addressing of applications with the pair values (app_name, host_name) and *(iv)* relying on broadcast and on managing message visibility with subscriptions to certain applications.

The Airplug software distribution is developed, mostly using Tcl/Tk programming language, to support several *modes* – independent implementations that are complementing each other. These modes include: the **terminal mode** – standard UNIX compatible command line implementation; the **emulation mode** – network emulator that simulates lower layers of protocol stacks while keeping upper layers the same as in experiments and the **live mode**, an efficient implementation suitable for execution on constrained embedded systems during real experiments. The Airplug software distribution is easily extended simply by writing applications (in any programming language) that follow those guidelines.

D. Software components

The application that makes the connection between the wireless sensors and the RSUs is called XBE. This application provides binary packet exchange between Xbee sensors and the system on which it is run (a PC or an embedded platform such as the Airbox). On the lower level, the XBE application enables communication and packet exchange in the proprietary format of Digi International using serial port and an Xbee module. On the upper layer the XBE application sends the temperature data gathered from the Xbee sensor.

XBE application can be used in different scenarios. While its basic feature is to gather data from wireless Xbee sensors and transfer them to the other applications, it can also be used to gather data in a predefined log file, to read data from log files that were created in previous experiments and to read user-generated data files. Basic manipulation of gathered data is possible. For example, we can offset to the data actually read to simulate different environmental conditions (useful for recreating a specific type of environment – *e.g.* winter temperatures during summer). Log files can be read at the same rate that was used during recording or at a higher rate as specified by application parameters. All these parameters can be used in both emulation and live modes [14].

For the purposes of testing and development of distributed data fusion algorithms we have developed the MET application. This application is a practical implementation of the belief function framework explained in Section II. This application is able to generate values, that are needed to study the robustness properties of the algorithm and to generate testing measurements according to a given function and periodicity. This last feature has been used on the Airbox units embedded in vehicles not equipped with temperature sensors and for emulation. Another possibility for this application is to take measurements from the other applications. This functionality is used in our experiments. XBE sends gathered data and MET applies data fusion on the received data (Figure 1).

MET can apply any user-defined frame of discernment (Ω set) and sigmoid functions defined for it. Independently of the data source, MET uses sigmoid functions, defined with equations (1) to (5) and calculates direct confidences according to the received values and the defined frame of

[1] http://www.digi.com/products/wireless-modems-peripherals/wireless-range-extenders-peripherals/xbee-sensors

[2] https://www.isee.biz/products/igep-processor-boards

discernment. It then proceeds with the calculation of the distributed confidence according to the algorithm detailed in Section II. These calculations are done periodically. The periods are independently defined for each instance of MET. When a new set of results is obtained the data is sent to the neighbors. One instance of the MET application handles input from one sensor, in case that we have more sensors we can run multiple instances of MET each handling one sensor input.

The applications described hereafter have been designed and tested using the terminal mode. Then we have deployed them on the previously described testbed using the Airplug live mode. The extensive study presented in the next section has been done by reproducing the testbed in the emulation mode. This is a convenient way to see the influence of parameters and input data.

IV. Experiments and results

A. Experimental setup

The testbed presented in the previous section has been validated during real tests. As we lack the space, we do not give details about these tests in this article. Instead, we focus on the behavior of the whole system with different inputs. For this purpose we rely on the emulator mode of the Airplug distribution. Airplug-emu accepts the same programs as those deployed on the Airboxes and it is able to reproduce with a high accuracy the real tests [1]. GPS positions are the same as in the real experiment, both for the RSUs and for the vehicle (Figure 3). GPS traces for vehicles have been saved during the real test and replayed during the emulation. There is no real sensor in the emulation; the XBE applications are also used in emulation mode: they output data coming from a log file recorded during the real test. Additionally, all the applications will also output the generated data, allowing us to study the behavior of our system

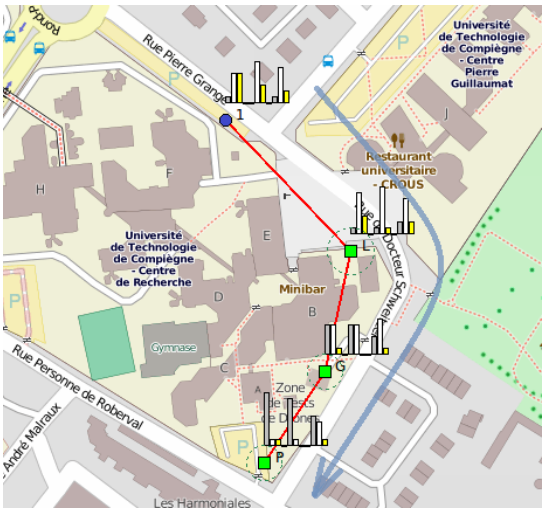


Fig. 3: Map showing positions of road-side units (green squares) and vehicle's (blue circle) direction of movement (blue arrow).

We divided our study in two groups of scenarios. The first one, called *static*, considers only the three RSU and their local temperature sensor (there is no vehicle). It allows us to show the influence of one RSU and its local measurements on its neighbors. The second group, called *dynamic*, considers the three RSUs with their sensors and a vehicle. Figure 3 shows a screenshot of the Airplug emulator, with openstreetmap tiles as background. It depicts the testbed close to the laboratory. The blue arrow shows the direction of the vehicle on the road, which is represented by a blue circle. The RSUs are represented by green squares. The vehicle first encounters the RSU-L (for Library), then the RSU-G (for Garage) and finally the RSU-P (for car Park). Each element in the emulation has a small histogram next to it, that represents the confidences computed in the previous step. The *dynamic* scenarios show the interest of our dangerous event detection system when a vehicle is approaching the icy road that is located close to RSU-P in our experiments.

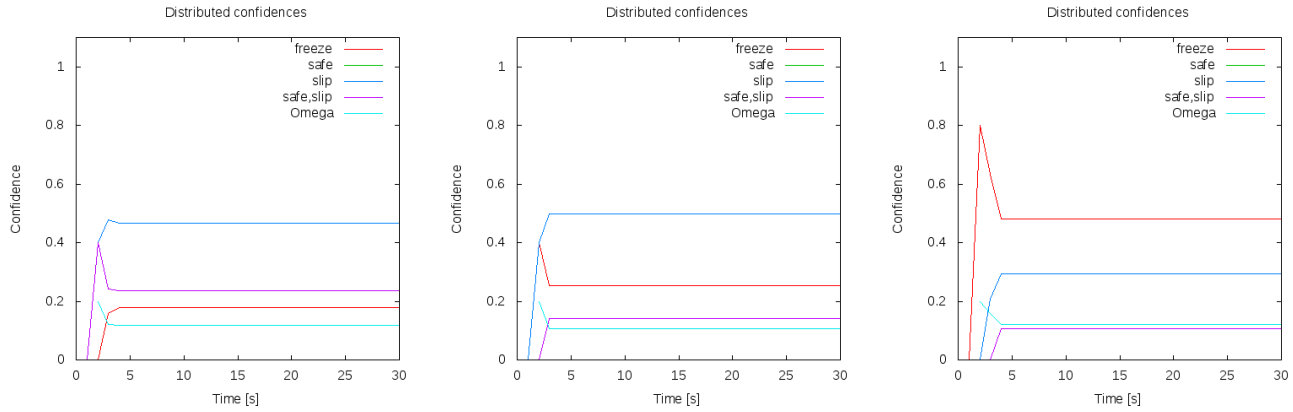
In each group of scenarios (static or dynamic), we considered two cases. The first one is the normal case where all sensors output correct temperature measurements. The second one is the disturbed case where the sensor connected to the RSU-G close to the garage is placed inside the garage. It then measures higher temperatures that do not reflect the road temperature. This is an extreme case of misplaced sensor. We would like to study the behavior of our system in such a situation. In the following, the first scenario is called *outside* because all sensors are placed outside. By opposition, the second is called *inside* because a sensor is placed inside of the garage. We will then study four scenarios, namely *static-outside*, *static-inside*, *dynamic-outside* and *dynamic-inside*.

Road temperatures in the outside scenarios are $3°C$, $-1°C$ and $-3°C$ for the RSU L, G and P respectively. The icy road representing the danger is close to the RSU-P. This setup is used to emulate sudden drops of temperature that may happen on the road due to changes in the atmospheric pressure or the environment. In the internal scenarios, the temperature given by the sensor of RSU-G is $21°C$. The other sensors output the same temperature as in the external scenario.

In order to have realistic emulated mobile scenarios, decreasing temperatures have been generated in the vehicle. For this purpose, we used the MET application in the vehicle in emulation mode so that it generates temperatures according to a given function. Starting temperature for the vehicle is $7°C$ and it decreases linearly each second with $0.133°C/s$ steps. The function for the temperature that the vehicle *"measures"* is constructed in such a way that the vehicle reports temperatures similar to those of the RSUs at the time it is passing close to them.

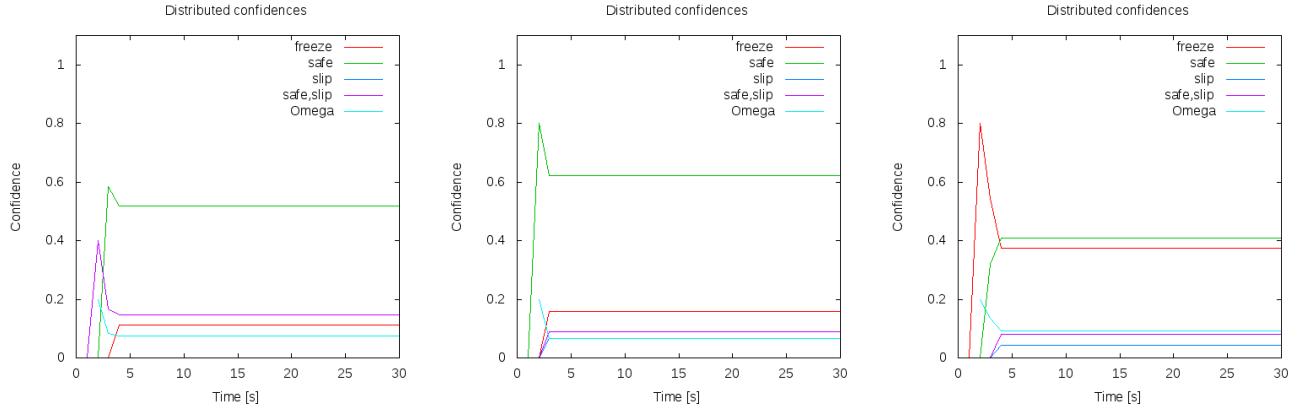
B. Results for the static scenarios

The static-outside scenario shows the adaptation of distributed confidences calculated by each RSU. Due to the fixed temperatures, chosen for the static-outside experiments, the distributed confidences (shown on Figures 4(a), 4(b) and 4(c)) have a short period of transition after which they settle to stable values. The values for distributed confidence correspond to their direct confidences (calculated from sigmoid functions knowing the temperature that each one of them is receiving) but we can also see the influence of the neighborhood on these calculations. Thus, for RSU-L, the distributed confidences for the $\{slip\}$ and $\{slip, safe\}$ subsets are different than the direct confidence that it calculates, and in this case it corresponds more to the values calculated by RSU-G.



(a) RSU L (b) RSU G (c) RSU P

Fig. 4: Distributed confidences for road-side units during the emulation in static-outside scenario.



(a) RSU L (b) RSU G (c) RSU P

Fig. 5: Distributed confidences for road-side units during the emulation in static-inside scenario.

The static-inside scenario shows how distributed data fusion adapts when one of the RSUs is receiving data that is very different compared to neighboring RSUs. In this case we can see, similarly to the previous case, the influence of the measurements from RSU-G on those of RSU-L. RSU-G measures the highest temperatures ($21°C$) thus it influences the increase of the distributed confidence for the $\{safe\}$ value. We can see the same influence on the RSU-P. This RSU is receiving measured temperature well beyond the level of its own $\{safe\}$ state and yet its $\{safe\}$ state has the highest value.

We used the static scenarios to show how different readings from external sources of data and direct confidence influence the distributed confidences of the neighbors. We can conclude that drastic changes in retrieved values on one RSU can bring in significant changes in the distributed confidences on the neighboring RSUs.

C. Results for the dynamic scenarios

In the beginning of both dynamic scenarios the vehicle is out of range from any RSU. The periods of communication with the different RSUs are given on Figure 6 and they are: T_L, the period in which the vehicle only communicates with RSU-L from $t = 12s$ to $t = 28s$; T_{LG}, the period in which the vehicle communicates with RSUs L and G from $t = 28s$ to $t = 41s$; T_{LGP}, the period in which the vehicle communicates with all three RSUs from $t = 41s$ to $t = 51s$; T_{GP} communication with RSUs G and P from $t = 51s$ to $t = 70s$; and T_P communication only with RSU P from $t = 70s$ to $t = 73s$.

For the dynamic-outside scenario we can see that the direct confidences for the vehicle (Figure 7(b)) are passing through all the values given by the sigmoid functions due to the linear change of temperature measured by the vehicle. The distributed confidence (Figure 7(c)) calculated by the vehicle is the most important result. It shows that the vehicle, as soon as it gets connected to RSU-L in the period T_L (Fig. 6), recalculates its

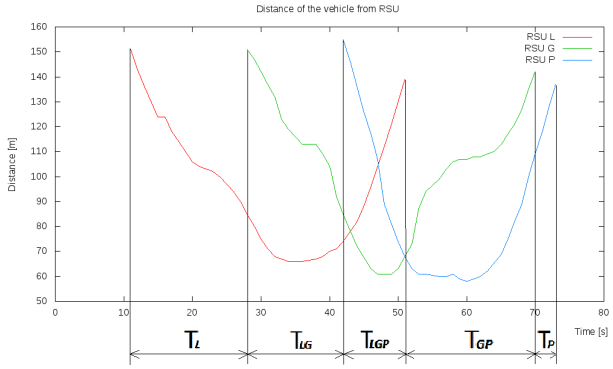
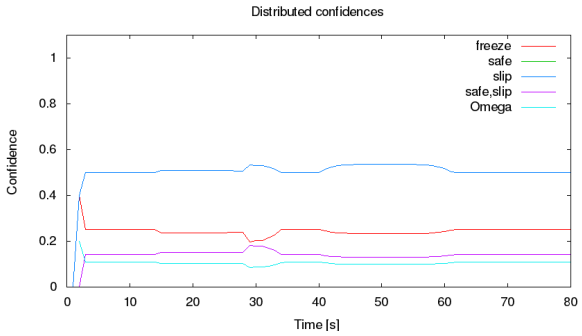
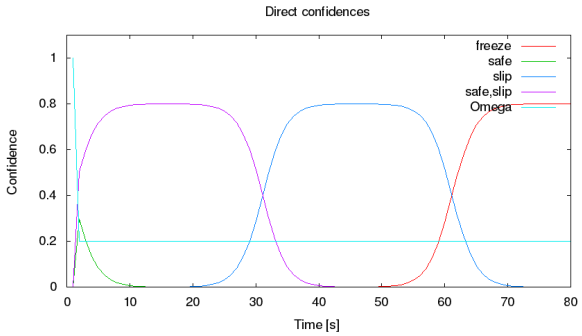


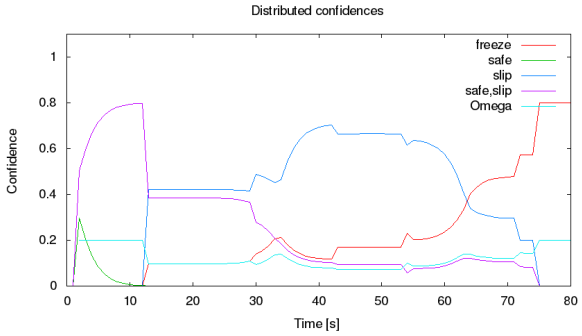
Fig. 6: Distances of vehicle from different RSUs; shown only when the vehicle is in the communication range of RSU



(a) Distributed confidences calculated by the RSU G. Influence of the measurement by the vehicles can be seen from $t = 28s$ to $t = 62s$

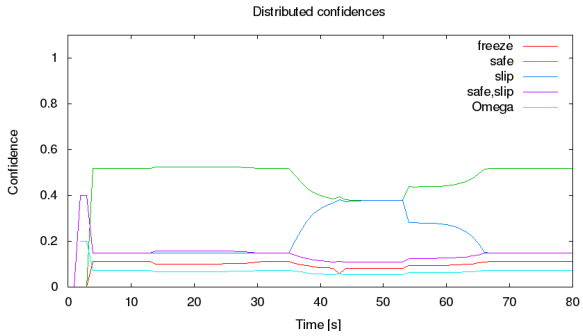


(b) Direct confidences calculated by the vehicle. Temperature is linearly decreasing from $7°C$ with the step of $-0.133°C$ and direct confidence is passing through all values given with sigmoid functions.

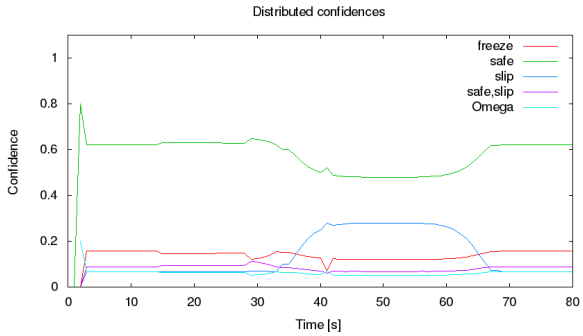


(c) Distributed confidences for vehicle in dynamic-outside scenario. Vehicle is getting warning at $t = 12s$ of the possibility of ice on the road before it actually reaches this point $t = 40s$

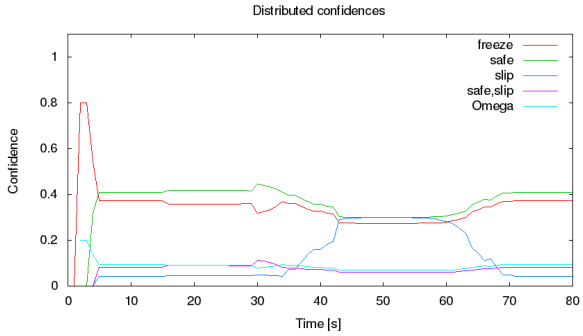
Fig. 7: Results for dynamic-outside scenario



(a) Distributed confidences calculated by the RSU L. Measures are adapted and they take into account direct confidences from vehicle.



(b) Distributed confidences calculated by the RSU G. Direct confidence calculations received from vehicle from $t = 33s$ to $t = 68s$ are influencing the change of distributed confidence calculated by RSU G.



(c) Distributed confidences calculated by the RSU P, two conflicting measures $\{freeze\}$ and $\{safe\}$ have highest values – due to the influence of RSU G.

Fig. 8: Results for dynamic-inside emulation scenario

distributed confidences giving the highest value to the $\{slip\}$ state under the influence of the confidences calculated by RSU-L. In this way the vehicle effectively predicts the possibility of a dangerous event (icy road) even though it is far away from it. The vehicle is then coming closer to the dangerous spot, the temperature measured by the vehicle decreases, changing its direct confidences. Comparing Figure 7(b) and Figure 7(c) we can deduce that in this period (T_{LG}), the distributed confidence change is **occurring before** its direct confidences change. In this case, the direct confidence just enhances this change, making it even more obvious. Similarly, we can deduce that the distributed confidence for the $\{freeze\}$ state is starting to increase much earlier than its direct confidence. This is again due to the influence of the measurements from the RSUs, and in this case this is prevalently due to the influence of RSU-L. A loss of connection, after the T_P period, brings back the distributed confidence to the level of the direct confidence calculated solely by the vehicle. In the Fig. 7(a) we can see in what way the direct confidence of the vehicle influences the distributed confidences of RSU-G, visible in the periods when the vehicle is connected to RSU-G (T_{LG}, T_{LGP} and T_{GP})

The main goal of the dynamic-inside scenario is to observe the way that distributed confidences are adapted in case of extreme differences in sensor readings. From the distributed confidences of each RSU (Figures 8(a)-8(c)), we can conclude that a high temperature that read by RSU-G has a large influence on both RSU-L and RSU-P. These three figures show that the influence of RSU-G dominates, bringing in the highest values for the distributed confidence for the $\{safe\}$ value. However, we can also observe that passage of the vehicle is

bringing in the change which corresponds to the real situation (decreasing temperatures in successive RSUs). This is visible in the periods T_{LGP} and T_{GP}, when the vehicle is connected to RSU-P and at the same time reaches lower temperatures in its own measurements. We can conclude that it influences the distributed confidences calculated by the RSUs bringing their calculations much closer.

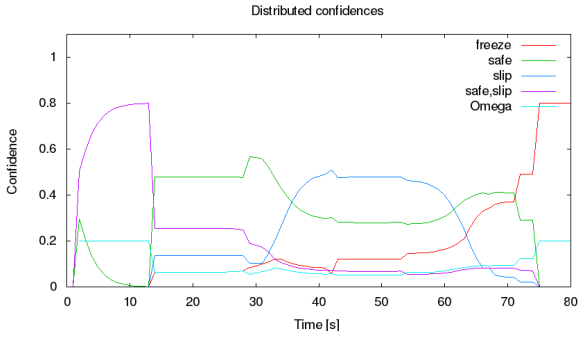


Fig. 9: Distributed confidences for the vehicle in the dynamic-inside scenario. Due to the readings of RSU-G, the vehicle is getting delayed warning of a possible icy road ahead.

The distributed confidence of the vehicle reveals that the measurements correspond to the real case *i.e.* direct confidences calculated from the vehicle are influenced by the RSUs. Nevertheless, the vehicle keeps its calculations of the distributed confidences, thanks to the belief function framework that spreads confidence to several subsets of the frame of discernment. In this case, the vehicle is again capable of detecting the dangerous spot on the road but due to the anomaly caused by the readings from RSU-G this detection is happening later when the vehicle gets in the range of all three RSUs and its own direct confidences start to change values.

V. Conclusion

In this paper, we have presented an application of distributed data fusion for early detection of dangerous events (icy roads), based on the belief functions theory. Our application combines several sensors measurements and propagates a mass vector with confidences on all the subsets of the frame of discernment that characterizes the different states of the road (freezing, slippery, safe). The main advantage of this technique is its robustness to wrong measurements and to give earlier warnings to drivers.

For development and testing purposes we have designed a complete testbed consisting of wireless sensors and solar RSUs communicating with WiFi between them and with vehicles. Extensive studies have been done by emulation, using data retrieved from our testbed, in different scenarios – with regular and erroneous measurements, using vehicles and RSUs. Results show that our application enables warning for approaching vehicles earlier than a simple alert broadcast generated when an average temperature is under a threshold. Moreover, data fusion can generate alerts for the approaching vehicles even when one of the sensors gives completely different measurements. Future work will include extensive road experiments with our testbed as well as adaptation of our application to other hazards. We also plan to investigate usage of pignistic probabilities in making the decision that is transmitted to the driver and to incorporate it in standardized environmental notification messaging system (DENM).

VI. Acknowledgments

This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02). It has been partially supported by the Celtic Plus project CoMoSeF Cooperative Mobility Service of the Future.

References

[1] A. Buisset, B. Ducourthial, F. El Ali, and S. Khalfallah. Vehicular networks emulation. In *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pages –, Suisse, August 2010.

[2] T. Denoeux. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artif. Intell.*, 172(2-3):234–264, 2008.

[3] F. Dressler and C. Sommer. On the impact of human driver behavior on intelligent transportation systems. In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5, May 2010.

[4] D. Dubois and H. Prade. Evidence, knowledge, and belief functions. *International Journal of Approximate Reasoning*, 6(3):295 – 319, 1992.

[5] B. Ducourthial, V. Cherfaoui, and T. Denoeux. Self-stabilizing distributed data fusion. In *Stabilization, Safety, and Security of Distributed Systems*, volume 7596, pages 148–162, Toronto, Canada, 2012.

[6] N.-E. El Faouzi, H. Leung, and A. P. Kurian. Data fusion in intelligent transportation systems: Progress and challenges - a survey. *Information Fusion*, 12(1):4–10, 2011.

[7] N. El Zoghby, V. Cherfaoui, B. Ducourthial, and T. Denoeux. Distributed data fusion for detecting sybil attacks in vanets. In *Belief Functions*, pages 351–358, 2012.

[8] M. Fogue, F. J. Martinez, P. Garrido, M Fiore, C. F. Chiasserini, C Casetti, J. C. Cano, C. M. T. Calafate, and P Manzoni. On the use of a cooperative neighbor position verification scheme to secure warning message dissemination in vanets. In *LCN*, pages 276–279, 2013.

[9] B. Khaleghi, A. M. Khamis, F. Karray, and S. N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

[10] R.C. Luo and M.G. Kay. Multisensor integration and fusion in intelligent systems. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(5):901–931, Sep 1989.

[11] R.C. Luo, Chih-Chen Yih, and Kuo-Lan Su. Multisensor fusion and integration: approaches, applications, and future research directions. *Sensors Journal, IEEE*, 2(2):107–119, Apr 2002.

[12] L. Malta, C. Miyajima, and K. Takeda. A study of driver behavior under potential threats in vehicle traffic. *Intelligent Transportation Systems, IEEE Transactions on*, 10(2):201–210, June 2009.

[13] F. Orhan and P.E. Eren. Road hazard detection and sharing with multimodal sensor analysis on smartphones. In *Next Generation Mobile Apps, Services and Technologies (NGMAST), 2013 Seventh International Conference on*, pages 56–61, Sept 2013.

[14] J. Radak, Bertrand D. Ducourthial, V. Cherfaoui, and S. Bonnet. Design and implementation of the vehicular network testbed using wireless sensors. In *The 8th International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN 2014)*, Benidorm, Spain, May 2014.

[15] B. Schweiger, C. Raubitschek, B. Bäker, and J. Schlichter. Elisatm - car to infrastructure communication in the field. *Comput. Netw.*, 55(14):3169–3178, October 2011.

[16] P. Smets. Data fusion in the transferable belief model. In *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, volume 1, pages PS21–PS33 vol.1, July 2000.

[17] P. Varaiya. Smart cars on smart roads: problems of control. *Automatic Control, IEEE Transactions on*, 38(2):195–207, Feb 1993.