Towards the use of multimedia contents to represent events in vehicular ad hoc networks

Nicolas Cenerario #1, Thierry Delot #2, Sergio Ilarri *3

#*LAMIH Laboratory, University of Valenciennes*
Le Mont Houy, 59313 Valenciennes - FRANCE
[1] `Nicolas.Cenerario@univ-valenciennes.fr`
[2] `Thierry.Delot@univ-valenciennes.fr`

**IIS Department, University of Zaragoza*
Maria de Luna 1, Zaragoza, 50018 - SPAIN
[3] `silarri@unizar.es`

Abstract:
VESPA (Vehicular Event Sharing with a mobile P2P Architecture)[1] is a system designed for vehicles to share information in inter-vehicle ad-hoc networks. The originality of VESPA is to support any type of event (e.g., available parking spaces, accidents, emergency brakings, obstacles in the road, information relative to the coordination of vehicles in emergency situations, etc.) in the network.

In this paper, we discuss the use of multimedia content to describe events and the impact of exchanging such data on the dissemination protocol used to diffuse the events to the potentially interested vehicles.

1 Introduction

Today, the car is indisputably the most heavily used mode of transportation. Unfortunately, its popularity has been accompanied by numerous problems, for example, in the areas of safety and the environment. In spite of significant efforts to reduce the number of persons dying on the road, this number remains quite high, mainly due to the human factor (e.g., accident-prone behavior or low response time). To reduce the number of accidents, a variety of programs, generally involving "Intelligent Transport Systems", have been initiated.

Thus, many works have focused on information exchange in vehicular ad hoc networks (VANETs). These wireless networks rely on the use of short-range networks (about a hundred meters), like IEEE 802.11 or Ultra Wide Band (UWB) standards for vehicles to communicate [LH05] and provide bandwidth in the range of Mbps. Using such communication networks, the driver of a car can receive information – for example, about accidents, traffic congestion or available parking spaces – from its neighbors.

These last years, different systems have been designed ([XOW04, MHD^{+}03, NDLI04,

[1] For more information, see: `http://www.univ-valenciennes.fr/ROI/SID/tdelot/vespa/`

FFH$^+$02, NDK04a]). They aim at assisting the drivers by providing them information about accidents, emergency brakings, or available parking spaces. VESPA follows a quite different approach. Contrary to other systems, dedicated to the dissemination of one particular type of information, the originality of VESPA is to support any type of event occurring on the roads, even mobile events. Indeed, numerous types of events –both mobile and stationary– are possible, since there is a lot of information that drivers may find relevant. For example, about accidents, traffic congestion, emergency braking situations, fuel prices, available parking spaces, emergency vehicles such as ambulances, obstacles in the road, or the behavior of drivers (e.g., strange manoeuvres due to intoxication or lack of vigilance), to name but a few possibilities. Therefore, VESPA relies on the concept of encounter probability to estimate the relevance of an event for a vehicle [DCI08]. VESPA also includes a dissemination protocol [CDI08]. This protocol ensures an adaptive broadcast of the events in the vehicular network according to their type.

In this paper, we present the basic functionalities of VESPA and focus on the event representation. We particularly highlight the interest of using multimedia data to enhance the description of events and facilitate the communication of relevant and useful information to drivers. The rest of this paper is organized as follows. Section 2 describes how the relevance of events received on a vehicle is estimated. In Section 3, we explain how the events occurring on the roads are represented to be communicated to potentially interested drivers. Section 4 focuses on our dissemination protocol in charge of that communication task. In Section 5, we discuss how to improve the basic description of events. We notably introduce the use of multimedia data to complete that description. Section 6 presents the architecture and the main functionalities of our VESPA prototype. Finally, Section 8 offers our conclusions.

2 Relevance estimation

Many pieces of information may be exchanged in the context of inter-vehicle communications, for instance to warn drivers when a potentially dangerous event arises (an accident, an emergency braking, an obstacle in the road, etc.) or to try to assist them (with information about available parking spaces, traffic congestions, real-time traffic conditions on a road, etc.). Those different events may be detected by a car and lead to the generation of a message transmitted to the other potentially interested vehicles, either directly or using multi-hop relaying techniques. Once received by a vehicle, the relevance of a message has to be evaluated, according to spatial and temporal criteria to determine whether the driver should be warned or the message should be further broadcast.

To estimate the relevance of an event (i.e., to determine whether a vehicle will encounter an event or not), it is necessary to have an estimation of the vehicle's trajectory. Therefore, VESPA exploits mobility and direction vectors to characterize the vehicle's displacement and so estimate a future position of the vehicle. These vectors are computed thanks to GPS position statements (including 3-dimensional coordinates as well as a statement of the GPS time) obtained regularly.

The current version of VESPA does not rely on the use of digital maps. Obviously, the information provided by such maps would provide very interesting information about the road network. This would clearly help in efficiently configuring the dissemination protocol. Thanks to such maps it would be possible to determine precisely the destination area of a message. For example, the last exit before reaching a traffic congestion could be determined as the minimum objective to reach while disseminating such an event. However, such maps are not always available or accessible on users' devices. So, our goal was initially to show the feasibility of the approach without using maps, even if we are now considering their use to improve our solution. Moreover, actual maps do not provide all the information needed to evaluate the relevance of the events received by a vehicle. For instance, such maps do not provide information about the entrances of parking lots whereas that information is crucial to determine the closest parking space for a vehicle about to reach its destination.

By using vectors, the estimated future position is highly dependent on the t_n and t_{n-i} time interval selected between the position statements used to compute the vectors. Thus, if t_n and t_{n-i} are far away, the estimation of the future position is not precise but provides an overall impression of the object's direction. If the time interval is shorter, then the estimation is much more precise on the short term but no global view of the displacement can be observed. As an example, see arrows A and B in Figure 1.

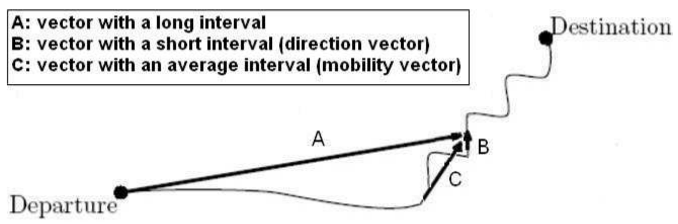

Figure 1: Mobility and direction vectors

Depending on the way we select the time interval $[t_{n-i}, t_n]$, we distinguish:

- The *direction vector*, which is computed with a short interval. It provides a quite precise estimated future position but only in the very short term (see arrow B in Figure 1).
- The *mobility vector*, whose role is to provide an overall impression of the object's

movement in addition to a good estimated future position. To achieve a good compromise between the previous two cases (arrows A and B in Figure 1), an "average" interval must be used to compute it (see arrow C in Figure 1).

Using the mobility and direction vectors and the positions of the vehicle and the event, we can deduce four elements which have an influence on the encounter probability:

- The minimal geographical distance between the vehicle and the event over time (Δd).
- The difference between the current time and the time when the vehicle will be closest to the event (Δt).
- The difference between the event's generation time (stored in *CurrentPosition*) and the moment when the vehicle will be closest to it (Δg, *expected age of the event*).
- The angle between the direction vectors of the vehicle and the event (denoted by a colinearity coefficient c).

As an example, Figure 2 shows the geometrical representation of Δd and Δt. In the figure, B represents the vehicle's position, C the position of a stationary event, and $\overrightarrow{AB}$ is the mobility vector of the vehicle. Point D can then be determined, which allows a right-angled triangle to be constructed in D with $[BC]$ as hypotenuse. D is the closest point to C on the straight line between A and B. $|\overrightarrow{DC}|$ (= Δd) represents the minimal geographical distance between the vehicle and the event over time. $|\overrightarrow{BD}|$ is the distance between the vehicle and the point D. Since the mobility vector $\overrightarrow{AB}$ has a temporal dimension, $|\overrightarrow{BD}|$ can be converted into time to obtain Δt.

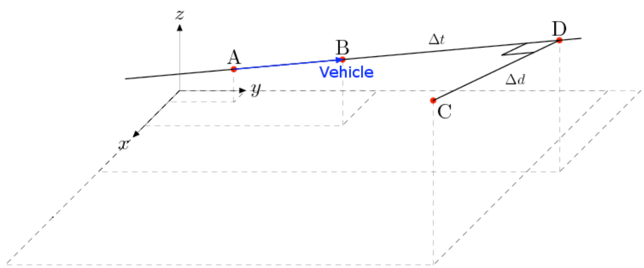

Figure 2: Representation of Δd and Δt

As explained previously, the vehicle estimates its direction vector and the event's direction vector. From these two direction vectors, a *colinearity coefficient* (c) is obtained, which is a measure of the angle formed by the vectors. For direction-dependent events that are not relevant for all nearby vehicles, but only for the vehicles travelling in a particular direction (e.g., an emergency braking, an accident, etc.), this allows us to determine whether the directions of the vehicle and the event match. For non-direction-dependent events, c is set to 0.

Once these Δd, Δt, Δg, and c values have been calculated, they are used to estimate an "encounter probability" between a vehicle and an event. The encounter probability (EP) is a value between 0% and 100%. It is computed, based on the previous values, using the following function:

$$\mathbf{EP} = \frac{\mathbf{100}}{\alpha \times \mathbf{\Delta d} + \beta \times \mathbf{\Delta t} + \gamma \times \mathbf{\Delta g} + \zeta \times \mathbf{c} + \mathbf{1}}$$

where α, β, γ and ζ are penalty coefficients with values ≥ 0. They are used to balance the relative importance of the Δd, Δt, Δg, and c values. The bigger the coefficient is, the more penalized the associated valued is when computing the EP. For example, the greater the α value, the shorter the spatial range where the event is relevant. β and γ are used so that only the information about events that will be encountered very rapidly and the most recent information is considered. Finally, ζ is used to weigh the importance of the colinearity coefficient. Notice that if the vehicle is moving away from the event, then Δt is 0 and Δd is the current distance to the event. Therefore, the computation of the EP makes sense even in cases where an interesting event (e.g., a parking space) is behind us. The EP is used to determine the relevance of an event. The greater its value, the more likely the vehicle is going to meet the event.

3 Representation of events

Thus far, existing V2V solutions have considered only a small subset of the possible types of events, primarily focusing on stationary events. However, numerous types of events – both mobile and stationary– are possible, since there is a lot of information that drivers may find relevant. For example, about accidents, traffic congestion, emergency braking situations, fuel prices, available parking spaces, emergency vehicles such as ambulances, obstacles on the road, or the behavior of drivers (e.g., strange maneuvers due to intoxication or lack of vigilance[2]), to name but a few possibilities. In order to determine the relevance of events, it is first necessary to classify the different types of events. In the rest of this section, we propose a system of event classification and describe how these events are represented in VESPA. For simplicity, not only all kind of events but also road hazards and available resources are called *events* in the following.

3.1 Event classification

The solution that we propose not only supports stationary events, such as the presence of gas stations, but also mobile events, such as an emergency vehicle asking preceding

[2]Lack of vigilance, or hypovigilance, can be detected today with oculometers using techniques that essentially count the driver's number of eye blinks.

vehicles to yield the right of way. When supporting such mobile events, the set of vehicles for which the event information is relevant evolves according to both the movements of the mobile event (in the example, the emergency vehicle) and the other vehicles involved (in the example, the preceding vehicles). Besides, the direction of traffic is also of major importance in establishing the relevance of shared information, even for non-mobile events (e.g., consider a traffic congestion affecting only the vehicles moving in one direction).

So, we classify inter-vehicle network events in four different categories:

1. *stationary, non-direction-dependent events*;
2. *stationary, direction-dependent events*;
3. *mobile, non-direction-dependent events*;
4. *mobile, direction-dependent events*.

By *direction-dependent events* we mean events that are not relevant for all nearby vehicles, but only for the vehicles traveling in a particular direction. On the other hand, *mobile events* are (as explained before) events whose locations change along time.

Let us illustrate our classification system by giving some examples. Available parking spaces correspond to stationary, non-direction-dependent events since they are static and may interest all vehicles close to that resource, regardless of the direction of movement. A warning about an accident is a stationary, direction-dependent event because its location is fixed and only those vehicles that are expected to encounter the accident will find the message relevant. The vehicles close to the accident but moving in the opposite traffic stream should ignore the message so as not to distract the driver and cause a second accident. Messages warning vehicles of the lack of vigilance of a person driving on a two-way road is a mobile, non-direction-dependent event because it concerns all vehicles likely to meet such driver, regardless of their direction of movement. Finally, an emergency vehicle broadcasting a message for other vehicles to yield the right of way is a mobile, direction-dependent event. Our goal in proposing such a classification of events is to support, in the same solution, all the types of events which can occur on the roads.

3.2 Basic Event Representation

In our solution, the four types of events identified in the previous section are used to represent all events occurring on the roads. In the following, we describe how these different events are represented when created[3] in order for them to be exchanged between vehicles (a summary of the attributes considered is shown in Table 1):

Each event is characterized by:

[3] We will not consider Human Machine Interface (HMI) aspects in this article. We rather focus on the representation and relevance estimation of events. The creation of those events may be initiated by devices embedded in the vehicles (for example by coupling the airbag system with the creation of an event representing an accident).

Attribute Name	**Type**
Key	string
Version	int
Importance	int
CurrentPosition	PositionAndTime
DirectionRefPosition	PositionAndTime
MobilityRefPosition	PositionAndTime
LastDiffuserPosition	PositionAndTime
HopNumber	int
Type	EventType

Table 1: Basic Event Representation

- A unique *Key*.
- A *Version* number to distinguish between different updates of the same event. Once generated, an event is disseminated among a set of potentially interested vehicles. To update the information transmitted to other vehicles, for example because a mobile event has moved, the vehicle which created the event may produce a new version of the same event.
- The *Importance* attribute, to determine whether the information should be presented to the driver or not. Unless the event is a very important one (e.g., an emergency braking), the driver is informed only if s/he is interested in that type of event.
- The *CurrentPosition* attribute indicating the generation time and place of the event.
- Two different preceding reference positions and their timestamps (*DirectionRefPosition* and *MobilityRefPosition*) for each vehicle to receive information to evaluate the mobility and direction of an event (see Section 2), which is necessary to estimate the event's relevance.
- The *LastDiffuserPosition* used by the dissemination protocol and containing the position of the last vehicle which relayed the message.
- The *HopNumber* attribute indicating the number of broadcasts of the message.
- The *Type* field describing more precisely the event considered (e.g., an accident, an emergency braking, etc.). This field is used to transmit concrete information to drivers when they need to be warned as we will see in Section 6.

4 Dissemination of events

Our objective as concerns the dissemination protocol is to disseminate different types of events (an accident, an emergency braking, an available parking slot, etc.) in the vehicular network. VESPA relies on vehicle-to-vehicle communication (V2V) (i.e., on spontaneous information exchanges between vehicles) and do not use mobile telephony networks (e.g., 3G), providing a worlwide access but increasing response time what is very penalizing in some situations (e.g., dealing with an emergency braking). Therefore, we have to support different dissemination modes according to the type of event considered. Indeed, an emergency braking has to be diffused to the vehicles driving in a particular direction whereas an available parking slot has to be transmitted to all close vehicles, whatever their direction, as it may interest them. VESPA uses a dissemination protocol[4] relying on the EP to determine the vehicles which have to broadcast some information they received. More precisely, when a message about an event is received by a vehicle, the vehicle will relay the message if and only if the value computed for the EP is greater than a predefined diffusion threshold. We indeed consider that a message relevant to a vehicle may also be relevant to its neighbors. Thus, our dissemination protocol allows diffusing the messages in the right direction, that is, towards the vehicles for which these messages may be relevant according to the type of event considered. Anyway, since this may happen at the same time on different vehicles, the same event may potentially be diffused numerous times by different vehicles. Therefore, to avoid flooding and so network congestion, our solution aims at desynchronizing the diffusions performed by the different vehicles. Thus, each vehicle waits for a period t before broadcasting the message. The size of that period depends on the distance between the receiving vehicle and the one which sent the message. The intuition behind this is to choose, among the neighboring vehicles which received the message, the farthest neighbor from the sender to relay the message. Indeed, this farthest neighbor may have the greatest number of neighboring vehicles not yet informed about the event being transmitted. It is so the best candidate to try to broadcast the message to all concerned vehicles as quickly as possible[5]. The value of t is determined by each vehicle as follows:

$$t = D \times (1 - \frac{d}{r})$$

where D is the maximum time to wait before broadcasting, r is the communication range of the wireless network used by the vehicles to communicate, and d corresponds to the distance between the receiving and the diffusing vehicle[6]. Since d may vary from 0 to r, t is between 0 and D.

5 Improving the description of events using multimedia content

Thanks to the basic attributes presented in Section 3, a driver can be informed or warned of events observed in its vicinity. Obviously, it can be interesting to add optional information

[4]See [CDI08] for more details.

[5]This removes the need of real-time monitoring the positions of the vehicles. Such a monitoring is indeed unrealistic in such dynamic environments.

[6]The value of d is computed using the position of the last vehicle that has relayed the event stored in the corresponding message.

to complete the description of each event, for example to indicate to the drivers the price of a parking or the length of a traffic congestion. Clearly, the use of multimedia content added to these event descriptions can also be interesting, either to give more details to the driver (e.g., a picture of an available parking space for the driver to know if s/he is able to park there) or to ease the delivery of the information to the driver (e.g., an audio message explaining that the warning transmitted to the driver corresponds to a dog walking on the road).

Multimedia content can help to communicate details about events to the drivers. Nevertheless, exchanging multimedia content in inter-vehicle networks is a challenging task. Indeed, whereas messages exchanged to described "classical" events (i.e., those without attachment of multimedia files) can be represented and disseminated over the network using a single packet, the size severely increases when multimedia files (e.g., pictures or audio files) are added to this message. Since vehicular networks are highly dynamic due to both the movements of the vehicles and the short range of the wireless communications, the exchange of multimedia contents between vehicles may fail frequently. This may be due to the fact that the available interaction time between two vehicles is not big enough (e.g., the vehicles are driving in the opposite direction) or due to a high probability of losing a packet.

So, we chose to use multimedia content only as optional attributes. Thus, even if the corresponding packets cannot be correctly received, the driver can be warned or informed of an event. Besides, when the workload is high in the vehicular network, the events can be relayed without their optional content in order to minimize the quantity of data transmitted and so to avoid losing packets.

We present in Table 2 the representation of an event including the optional attributes. An optional attribute is described by its name (represented as a string), a type and a value (several triples are allowed, as we will show later in Table 4). If the type of the optional attribute is different from "text" (e.g., "audio", "picture", etc.), then the associated value corresponds to an identifier referencing another entity where the raw data is stored (see Table 5).

Attribute Name	**Type**
Key	string
Version	int
Importance	int
CurrentPosition	PositionAndTime
DirectionRefPosition	PositionAndTime
MobilityRefPosition	PositionAndTime
LastDiffuserPosition	PositionAndTime
HopNumber	int
EventType	string
OptionalContent	$<$ string , type , string $>$

Table 2: Event representation with optional content

Attribute Name	Type
IdContent	string
Data	byte []

Table 3: Multimedia content storage

To illustrate the usage of this structure, an example is proposed in Tables 4 and 5.

Attribute Name	Example
Key	v1e1
Version	1
Importance	1
CurrentPosition	50°19’15.91 N 3°30’51.11 E 10h25m17s
DirectionRefPosition	null
MobilityRefPosition	null
LastDiffuserPosition	50°19’15.91 N 3°30’51.11 E 10h25m17s
HopNumber	1
EventType	parking
OptionalContent	< “Description” , text , “Available parking space” “For disabled” , text , “No” “Cost” , text , ”2 Euros / h” “Picture” , picture, “idpics01” >

Table 4: Parking example

Attribute Name	Example
IdContent	idpics01
Data	{0001 ... 0110}

Table 5: Parking multimedia content example

6 VESPA Prototype

In this section, we describe our prototype. We first present basic aspects of the prototype and then describe its software architecture.

Figure 3: Testing VESPA in a real environment

6.1 General Presentation

For obvious scalability reasons, our dissemination and relevance estimation techniques were evaluated on a simulator. Anyway, a prototype of VESPA has been implemented using Microsoft .Net/C# to observe its behavior in "real conditions"[7]. Our prototype was used to validate the dissemination of the events according to their Encounter Probability. It supports different events such as a vehicle leaving a parking space (i.e., an event relevant for all the vehicles that are close to that space during a given period of time), and an emergency braking (only relevant for the vehicles following the vehicles generating that event).

Our VESPA prototype runs on PDAs equipped with embedded GPS receivers. The dissemination protocol presented in Section 4 relies on Wi-fi communications to support the exchanges between the vehicles. Using our prototype, a driver can receive on her/his PDA information about events transmitted by neighboring vehicles. As described in Figure 3, s/he can basically observe the type of event (e.g., an available parking space, an accident, etc.), the distance between his/her car and the event, and an arrow indicating the direction to follow to reach the event.

In Figure 4, we present an example describing how a driver can access the optional information attached to an event in the case of a parking space event. The first screen presents our prototype interface when waiting for potential events. The second screen shows the basic information printed on reception of an available parking space event. The driver can also watch the optional attributes, which may correspond to multimedia data such as a picture in our example. As concerns the exchange of multimedia data through the vehicular ad hoc networks, we only manage to exchange small files (i.e., a few KBytes). Indeed, we do not use any antenna with the smartphones yet and the communication range is about 100 m, which limits the duration of the exchanges between the vehicles, in particular when they move in different directions.

[7]The number of vehicles used during our field tests remain limited for the moment.

Figure 4: Information about an available parking space event

Let us note that in real conditions the generation of many events could be initiated using the numerous sensors embedded in our cars (for example, by coupling the airbag system with the creation of an event representing an accident). Since the smartphones used are not yet connected to these sensors (i.e., to the CAN bus of the vehicle), the generation of the events is managed by the device (using the GPS signal) in the current version of our prototype.

6.2 Architecture

The architecture of VESPA, which is deployed on every equipped vehicle, is presented in Figure 5, where the following main elements can be distinguished:

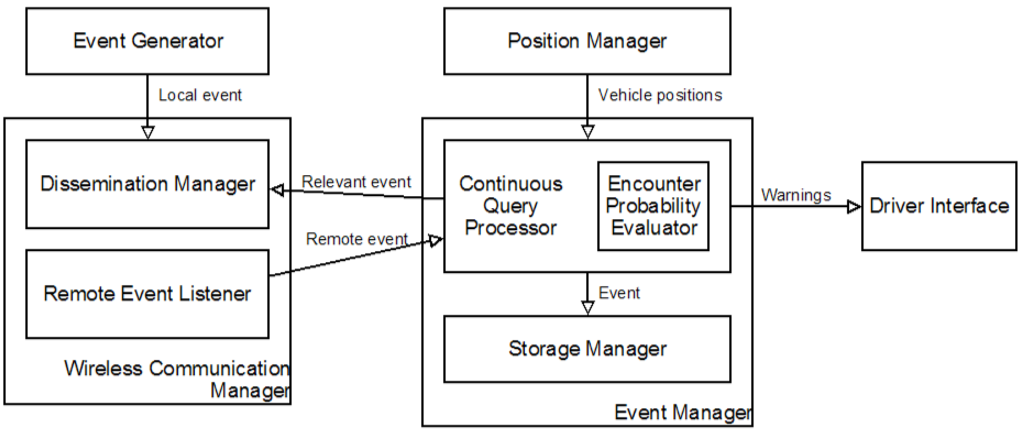

Figure 5: VESPA architecture

- The *Wireless Communication Manager* is in charge of the reception and transmission of events. This module is composed by the *Dissemination Manager*, which allows the vehicle to broadcast events, and the *Remote Event Listener*, in charge of the reception of the events transmitted by the neighboring vehicles.

- The *Event Manager* handles the events received by the vehicle. It is composed of the *Continuous Query Processor*, which processes the active continuous queries to determine not only whether the vehicle is going to encounter the event or not (by using an *Encounter Probability Evaluator*) but also if the driver is interested or not in that event. Finally, the *Storage Manager* is in charge of deciding about the storage and removal of events.
- The *Driver Interface* is the graphical user interface used to interact with the driver (e.g., showing information about events detected).
- The *Position Manager* interacts with the GPS receiver of the vehicle to retrieve information regarding the location of the vehicle.
- Finally, the *Event Generator* releases events detected by the vehicle. The generation of many events could be initiated using the numerous sensors embedded in modern cars (for example, by coupling the airbag system with the creation of an event representing an accident) or via other static information sources (e.g., sensors on a road). Specific Human Machine Interface (HMI) aspects are not considered in this article.

In the following, we briefly explain the way the different modules interact:

1. An event is first received by the Continuous Query Processor. The Encounter Probability Evaluator computes the encounter probability of the event using the information provided by the Position Manager. The relevance of an event may change continuously due to the different dynamic factors affecting the computation of the encounter probability, such as the distance to the event (as explained in Section 2). Therefore, the Continuous Query Processor, using the Encounter Probability Evaluator, evaluates periodically the active continuous queries to verify which events must be reported to the driver through the Driver Interface. Moreover, a new event received could also be processed immediately if its *Importance* field has a high value (e.g., for accidents or emergency braking situations). Each event for which the encounter probability is higher than a *relevance threshold* must be checked against the set of active continuous queries.
2. The Storage Manager is informed by the Encounter Probability Evaluator about the encounter probabilities it computes. If the encounter probability of a previously stored event is smaller than a *storage threshold*, then the Storage Manager removes the event from the storage area. On the contrary, if the encounter probability of a new event is greater than the storage threshold, the event is stored.
3. For a new event received, in case its encounter probability is higher than the diffusion threshold, the Dissemination Manager is contacted by the Continuous Query Processor to broadcast the event and inform other vehicles.

7 Related Works

Numerous works have addressed communication protocols in vehicular networks, either in the context of geocasting protocols (for example, [NI97, KV99]), whose goal is to transmit data to all the targets within an area, or in the context of dissemination protocols [NSI06, FMH+02, LSCM07, XOW04, NDK04b].

However, those works consider very small messages exchanged between vehicles not compatible with the use of multimedia data. Up to our knowledge, the only work that considers multimedia data in vehicular networks is [GAZ05], where an architecture to provide live video streaming to vehicles is presented (*V3*). Vehicles within an interesting region are assumed to be able to capture video data from that region. In this approach, two areas are considered: the *trigger message forwarding zone* (*TMFZone*, which is the area within which a query must be forwarded) and the *data forwarding zone* (*DFZ*, which is the area from where video data must be forwarded to the interested vehicle). Both the case where there is a single receiving vehicle and the case with multiple receivers (where some optimizations are possible by using multicast, in order to reduce sending duplicate packets) are considered. A store-carry-and-forward approach is proposed, and several algorithms are analyzed by considering the tradeoff between transmission delay and bandwidth overhead. For example, one of the algorithms proposed to select a data forwarder is the *optimal selection* where, assuming that each vehicle knows the mobility function of other vehicles within its communication range, the vehicle that could reach the receiver (or the next network partition) sooner is selected.

8 Conclusion & Perspectives

In this paper, we have presented the basic functionalities of VESPA and discussed improvements in the representation of events. Particularly, we have highlighted the interest of using multimedia data. VESPA can be seen as a system complementary to existing navigation systems. Indeed, whereas navigation systems can be used to guide drivers and show them the location of different points of interest (e.g., petrol stations, railway stations, airports, etc.), VESPA can provide them ephemeral information about the road hazards they may encounter along their displacement (e.g., information about an emergency braking, an available parking space, etc.).

Our current work is to evaluate VESPA in real conditions using our prototype. We are also studying how to improve it. Therefore, we are working on the aggregation of the events received by a vehicle. Our goal is to extract additional knowledge to be used by the drivers. For example, using the summaries generated with the available parking spaces [DDM+08], it becomes possible to determine the areas where the probability to find an available parking space is high.

Acknowledgements

The present research work has been supported by the International Campus on Safety and Intermodality in Transportation the Nord-Pas-de-Calais Region, the European Community, the Regional Delegation for Research and Technology, the Ministry of Higher Education and Research, and the National Center for Scientific Research. The authors gratefully acknowledge the support of these institutions.

References

[CDI08] N. Cenerario, T. Delot und S. Ilarri. Dissemination of information in inter-vehicle ad hoc networks. In *Intelligent Vehicles Symposium (IV'08)*, 2008.

[DCI08] T. Delot, N. Cenerario und S. Ilarri. Estimating the relevance of information in inter-vehicle ad hoc networks. In *IEEE International Conference on Mobile Data Management (MDM'08) - Workshops*, 2008.

[DDM+08] B. Defude, T. Delot, J.L. Zechinelli Martini, N. Cenerario und S. Ilarri. Extraction de connaissances dans les réseaux ad hoc inter-véhicules. In *ACM conf. on Mobilité & Ubiquité (Ubimob'08)*, May 2008.

[FFH+02] A. Festag, H. Füßler, H. Hartenstein, A. Sarma und R. Schmitz. Fleetnet: Bringing car-to-car communication into the realworld. In *World Congress on Intelligent Transport Systems (ITS)*, 2002.

[FMH+02] H. Füβler, M. Mauve, H. Hartenstein, M. Käsemann und D. Vollmer. A Comparison of Routing Strategies for Vehicular Ad Hoc Networks. Bericht TR-02-003, Department of Computer Science, University of Mannheim, July 2002.

[GAZ05] Meng Guo, Mostafa H. Ammar und Ellen W. Zegura. V3: A vehicle-to-vehicle live video streaming architecture. *Pervasive and Mobile Computing*, 1(4):404–424, December 2005.

[KV99] Y.-B. Ko und N. H. Vaidya. Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms. In *Second IEEE Workshop on Mobile Computer Systems and Applications (WMCSA'99)*, Seite 101. IEEE Computer Society, October 1999.

[LH05] J. Luo und J-P. Hubaux. A Survey of Research in Inter-Vehicle Communications. In *Embedded Security in Cars - Securing Current and Future Automotive IT Applications*. Springer-Verlag, 2005.

[LSCM07] C. Lochert, B. Scheuermann, M. Caliskan und M. Mauve. The Feasibility of Information Dissemination in Vehicular Ad-Hoc Networks. In *Conf. on Wireless On demand Network Systems and Services (WONS'07)*, 2007.

[MHD+03] P. Morsink, R. Hallouzi, I. Dagli, C. Cseh, L. Schafers, M. Nelisse und D. De Bruin. CARTALK 2000: Development of a cooperative ADAS based on vehicle-to-vehicle communication. In *Intelligent Transport Systems and Services*, 2003.

[NDK04a] S. Nittel, M. Duckham und L. Kulik. Information Dissemination in Mobile Ad-hoc Geosensor Networks. In *3rd Int. Conf. on Geographic Information Science*, 2004.

[NDK04b] S. Nittel, M. Duckham und L. Kulik. Information Dissemination in Mobile Ad-Hoc Geosensor Networks. In *3rd International Conference on Geographic Information Science (GIScience'04)*, Seiten 206–222, 2004.

[NDLI04] T. Nadeem, S. Dashtinezhad, C. Liao und L. Iftode. TrafficView: Traffic Data Dissemination Using Car-to-Car Communication. *ACM Sigmobile Mobile Computing and Communications Review, Special Issue on Mobile Data Management*, 8(3):6–19, 2004.

[NI97] J. C. Navas und T. Imielinski. GeoCast – geographic addressing and routing. In *Third Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'97)*, Seiten 66–76. ACM, September 1997.

[NSI06] T. Nadeem, P. Shankar und L. Iftode. A Comparative Study of Data Dissemination Models for VANETs. In *3rd Int. Conf. on Mobile and Ubiquitous Systems (MOBIQUITOUS'06) - Workshops*, 2006.

[XOW04] B. Xu, A. M. Ouksel und O. Wolfson. Opportunistic Resource Exchange in Inter-Vehicle Ad-Hoc Networks. In *5th Int. Conf. on Mobile Data Management*, 2004.