A Cooperative Reservation Protocol for Parking Spaces in Vehicular Ad Hoc Networks

Thierry Delot
Univ Lille Nord de France
UMR UVHC/CNRS 8530
Valenciennes, France
tdelot@univ-valenciennes.fr

Nicolas Cenerario
Univ Lille Nord de France
UMR UVHC/CNRS 8530
Valenciennes, France
ncenerar@univ-valenciennes.fr

Sergio Ilarri
IIS Department
Univ of Zaragoza
Zaragoza, Spain
silarri@unizar.es

Sylvain Lecomte
Univ Lille Nord de France
UMR UVHC/CNRS 8530
Valenciennes, France
slecomte@univ-valenciennes.fr

ABSTRACT

By exchanging events in a vehicular ad hoc network (VANET), drivers can receive interesting information while driving. For example, they can be informed of available parking spaces in their vicinity. Obviously, a suitable protocol is needed to disseminate the events efficiently within the area where they are relevant. Moreover, in such a competitive context where each vehicle may be interested in the resource, it is crucial not to broadcast that resource to each driver in the vicinity. Otherwise, those drivers would waste time trying to reach a parking space and only one of them would be satisfied, which would lead to a poor satisfaction in the system.

To solve this problem, we detail in this paper a reservation protocol that efficiently allocates parking spaces in vehicular ad hoc networks and avoids the competition among the vehicles. We have integrated our protocol within VESPA, a system we have designed for vehicles to share information in VANETs. An experimental evaluation is provided, which proves the usefulness and benefits of our reservation protocol.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Distributed systems—*distributed data sharing, data management in vehicular networks*; H.4.m [**Information Systems Applications**]: Miscellaneous—*applications for vehicular networks*

General Terms

Algorithms, Design, Performance, Experimentation

©ACM, (2009). This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Mobility'09: Proceedings of the 6th International Conference on Mobile Technology, Application & Systems, Sep 2-4, Nice, France, 2009. http://doi.acm.org/10.1145/1710035.1710065
.

Keywords

Vehicular networks, data sharing, parking space allocation

1. INTRODUCTION

Nowadays, there is a great interest in developing systems to assist drivers on the road, providing them with different types of relevant information. VANETs 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 [9] and provide bandwidth in the range of Mbps. Using such communication networks, the driver of a car can receive information from its neighbors. Many pieces of information can 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 on the road, etc.) or to try to assist them (with information about traffic congestions, real-time traffic conditions, etc.). Particularly, different works have addressed the advertisement of available parking spaces [11, 3, 10, 2]. Finding an available parking space is indeed stressful, time-consuming and contributes to increasing traffic. Besides, it leads to fuel consumption and environment pollution due to the emission of gases. Some works, such as [3], emphasize the costs of searching for parking spaces. According to that work, nearly one out of two vehicles on the move are searching for a parking space.

The work presented in this paper is an extension of VESPA[1] (Vehicular Event Sharing with a mobile P2P Architecture), which is a system developed to share information about events in inter-vehicle ad hoc networks. In such environments, data is received from other vehicles and stored locally in a data cache. Then, query evaluation techniques are used to sift through the stored information to determine what is relevant for that time and location, and issue a warning or transmit information to the driver when necessary. Data about the events occurring on the road or available resources such as parking spaces have to be communicated to a potentially large set of vehicles, depending

---

[1] See `http://www.univ-valenciennes.fr/ROI/SID/tdelot/vespa/` for more information.

on the relevance of the data to the drivers. As opposed to other proposals, VESPA aims at supporting all types of events. Thus, VESPA proposes a dissemination protocol [4] and a relevance estimation mechanism [7] not only suitable for stationary events (e.g., an emergency braking, an accident, etc.) but also for mobile events (e.g., an emergency vehicle asking preceding vehicles to yield the right of way, a vehicle with a non-functioning brake light, etc.). When supporting such mobile events, the set of vehicles for which the event information is relevant evolves according to both the movements of the vehicle generating the event (e.g., a vehicle that brakes suddenly) and the other vehicles involved (in the previous example, the vehicles behind). 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 jam affecting only the vehicles moving in one direction).

In this paper, we focus on the exchange of information about available parking spaces using VESPA. As opposed to other types of events, it is not enough to indicate the presence of the event to the driver. Indeed, if the same information (i.e., the same available parking space) is presented to several interested drivers, this will lead to a competition between the vehicles and only one of them will be able to take that parking space. It is so crucial to propose a solution for parking spaces to be “reserved” and so communicated to a single driver. The fully decentralized environment imposed by vehicular networks makes that reservation process particularly challenging since vehicles keep moving and no reliable link or central server can be used. Although other solutions have been proposed to disseminate information about available parking spaces using short range communications (e.g., [11]), to the best of our knowledge, no other work has tackled the problem of parking space reservation in VANETs.

The structure of the rest of this paper is as follows. In Section 2, we introduce the representation of events in VESPA and describe how the relevance of parking spaces received by vehicles is evaluated using the concept of Encounter Probability. In Section 3, we present our reservation algorithm. In Section 4, we evaluate experimentally our solution. In Section 5, we compare our work with other approaches. Finally, in Section 6 we summarize our conclusions and indicate some ideas for future research.

2. RELEVANCE OF EVENTS

One of the major problems in inter-vehicle applications is how to estimate the relevance of the events received. The classical approach is to define a relevance function used to determine whether an event received should be considered or ignored. The relevance function used in VESPA, called *Encounter Probability* (*EP*), is used to verify whether an event is relevant for a vehicle or not. Thus, when an event is received by a vehicle, the EP for this event is computed. If the EP reaches a certain threshold, the event is relevant and has to be communicated to the driver if s/he is interested in that type of event.

In the following, we first describe how events are represented and then present the computation of the *Encounter Probability* (*EP*).

2.1 Representation of Events in VESPA

VESPA relies on a generic structure to represent events whatever their type. Each event is described using the following attributes: 1) A unique *Key* set by the event source, 2) a *Version* number allows to distinguish between different updates of the same event (e.g., used to indicate that a mobile event has changed its location), 3) an *Importance* value helps to determine the urgency of presenting that information to the driver (e.g., accidents that may affect the driver have a high importance), 4) a *CurrentPosition* indicates the time and place that the data was generated, 5) a *DirectionRefPosition* and a *MobilityRefPosition* store preceding reference positions used to compute an Encounter Probability (see Section 2), 6) a *LastDiffuserPosition* and a *HopNumber* contain the location of the last vehicle which relayed the message and the number of rediffusions of the event (for purposes of the dissemination protocol, as explained in [4]), and 7) a *Description* field contains additional information for the driver.

The type of the event (stationary or mobile, direction-dependent or not) is not explicitly represented as an attribute of the event, as it can be inferred from some of the other message fields. Thus, when dealing with a stationary object/event, the *MobilityRefPosition* will always be set to *null*. Similarly, for non-direction-dependent events the value of *DirectionRefPosition* will be set to *null* to allow the identification of such type of event.

2.2 Computation of the EP

In order to compute the Encounter Probability between a vehicle and an event, two movement vectors are defined for a vehicle (computed by sampling the vehicle's locations periodically): the direction vector and the mobility vector. The *direction vector* allows estimating future positions of the vehicle on a short term, whereas the *mobility vector* captures an overall impression of the vehicle's direction and allows to estimate future positions on the long term. Each vehicle can compute its direction vector and its mobility vector easily. Similarly, each vehicle can compute the *mobility and direction vectors of the events* it receives. For that purpose, it uses the data associated to the events, and more precisely the *CurrentPosition* attribute and either the *DirectionRefPosition* or the *MobilityRefPosition* attribute, respectively. Finally, for each event, a *vehicle's mobility vector (and direction vector) in relation to the event* is computed by the vehicle, so that managing a single mobility and direction vector for each couple <vehicle, event> is enough.

The vehicle's mobility vector and direction vector in relation to the event are used to compute four elements (an example of the first two elements, Δd and Δt, is shown in Figure 1, where B represents the vehicle's position, C the event's position, and $\overrightarrow{AB}$ is the mobility vector of the vehicle in relation to the event):

- 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 time when the event is generated and the moment when the vehicle will be closest to the event (Δg).
- The angle between the vehicle's direction vector and the event's direction vector (represented by a colinearity coefficient c).

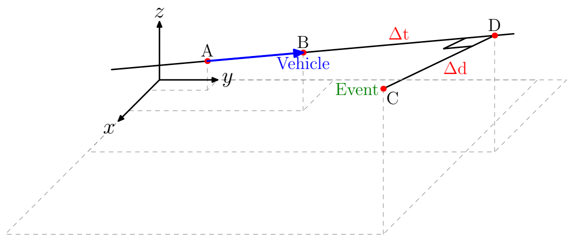

Figure 1: Geometrical representation of Δd and Δt for a sample stationary event

Once these values have been calculated, they are used to estimate an "Encounter Probability" (EP, a value in the range of 0% to 100%) between the vehicle and the event:

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

where α, β, γ and ζ are penalty coefficients with values ≥ 0, and Δt and Δg are expressed for convenience in seconds. 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 encounter probability. 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 most recent information and the information about events that will be encountered very rapidly 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 (i.e., Δd is no more the minimal geographical distance between the vehicle and the event over time in that case). Therefore, the computation of the EP makes sense even in cases where an interesting event (e.g., a parking space) is behind the vehicle. 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. For more details about the computation of the Encounter Probability, along with an evaluation of its benefits, we refer the interesting reader to [4].

The dissemination protocol described in [4] which aims at delivering events to potentially interested vehicles also relies on the EP mechanism. Indeed, the probability that an event relevant for a vehicle is also relevant for its neighbors is high. So, when the EP computed for the event reaches a threshold, the dissemination of the event has to be continued in the network. Such an EP-based dissemination protocol ensures the adaptivity of the dissemination of an event according to its type. This is indeed crucial when several types of events are considered (e.g., the information about an available parking space has to be broadcasted to all surrounding vehicles whereas the information about an emergency braking should only be diffused to the vehicles behind).

2.3 EP-Based Continuous Event Processing

The Encounter Probability mechanism allows to determine the relevance of an event for a vehicle. The greater the probability, the more likely the vehicle is going to encounter the event. Nevertheless, it is also crucial to take the driver's preferences into account in order to select which events should be presented to the driver. Indeed, whereas the events corresponding to critical situations (e.g., an emergency braking), identified with a high value for the *Importance* attribute (see Section 2.1), should always be communicated to the driver, s/he should not be disturbed by unwanted information while driving. For instance, s/he should not receive information about available parking spaces unless s/he specified her/his interest in that type of event.

Thus, a distinction between events that are *relevant for the vehicle* and events that are *relevant for the driver* is made. An event is relevant for the vehicle if the vehicle will probably encounter the event. An event is relevant for the driver if it is relevant for her/his vehicle and besides the driver is interested in that type of event.

Obviously, the values of the Encounter Probability for events constantly change because the vehicle (and sometimes also the events) move. So, every T time units these changes are tracked by the vehicle in order either to invalidate a warning (e.g., for example because the vehicle has changed its direction and will finally not encounter the event) or to react to a change in the driver's preferences (e.g., the vehicle is about to reach its final destination and the driver needs now to find an available parking space). These changes in the Encounter Probability may also lead to changes in the set of potential events that must be shown to the driver. For instance, when dealing with parking spaces, new appearing empty slots have to be added to the list presented to the driver whereas the items which become obsolete have to be removed.

To present only relevant information to the driver, the driver's interests are stored in a vehicle in the form of queries which are re-evaluated continuously, by a *Continuous Query Processor* module, to retrieve relevant events. At least one continuous (implicit) "background" query is used to detect all the events relative to dangerous situations which have to generate a warning for the driver (events characterized by a high value in the *Importance* attribute). This query continuously retrieves the events with both a high encounter probability and a high value of the *Importance* field. The information extracted by executing the query is then used to warn the driver. Other continuous queries may be evaluated concurrently to satisfy the driver's needs. For instance, a query could be in charge of retrieving relevant parking spaces, that is, events whose EP exceeds the relevance threshold and whose description field is "Available Parking Space". In the experiments that we have performed in a real environment, the query processing overhead is not high.

Along this section 2, we have described how to estimate the relevance of events both at the vehicle's and driver's levels. For most events, these mechanisms can be used efficiently to disseminate the events in the vehicular network to warn drivers. However, disseminating information about an available parking space with the basic VESPA approach is not enough, as this would lead to competition between vehicles to try to take that space. In the next section, we present a solution to allocate available parking spaces in VANETs.

3. RESERVATION PROTOCOL

Coordinating different vehicles in vehicular ad hoc networks for them to choose one vehicle to which the parking space will be allocated is not an easy task. Indeed, no centralized server is available in such environments where vehicles only communicate through short range communication networks. Moreover, all the vehicles have the same importance/role. So, we propose in the following a protocol in which a *coordinator* vehicle is chosen for each parking space. The role of such a coordinator is to collect, from its neighbors interested in finding an available parking space, the necessary information to decide to which vehicle the resource will be allocated. In the following, we will call "reservation" the process consisting of allocating parking spaces to vehicles. Our goal is to ensure that the information about a parking space is shown only to the driver of the vehicle that is chosen to occupy it, in order to minimize competition. Anyway, even if we try to help the elected vehicle to reach the parking space while it is still available, we cannot ensure that no other driver will see and use the available parking space before. This is obviously unavoidable. What our protocol eventually ensures is an effective dissemination of information about available parking spaces, without leading to a competition.

3.1 General Principle

Using vehicle-to-vehicle communications, a vehicle can inform the other surrounding ones when it is about to leave a parking space. Therefore, a message describing the event "available parking space" is generated and broadcasted using VESPA, as explained before. To avoid causing an unnecessary competition among vehicles, a suitable reservation protocol should satisfy the following properties:

- indicate to the driver an available parking space reserved for her/his vehicle (and not a list of all the received available parking spaces);
- maximize the probability that the parking space is still available when the driver arrives there;
- avoid that a parking space remains available if there are vehicles searching in its vicinity (i.e., maximize the use of the resources);
- be equitable (i.e., ensure that a vehicle has not a higher priority than the others);
- minimize the actions to be performed by the driver, with two goals: 1) not to disturb her/him while driving, and 2) to preserve the fairness of the protocol by not allowing the drivers to disseminate themselves misleading information that they could use to their own advantage (i.e., to obtain a parking space before the others).
- avoid network congestion.

Our solution to reserve parking spaces relies on the election of one *coordinator* per available parking space (i.e., a vehicle with a temporary particular role in the allocation of an available space). The coordinator is responsible for the allocation of the parking space to a vehicle according to a predefined policy. In the following, we detail how our protocol works. We first consider the case where there are vehicles interested in the resource in the communication range of the vehicle diffusing it.

At first, the coordinator is the vehicle that leaves the parking space[2]. It sends a message to inform the vehicles in its communication range that a parking space is available. Then it waits for potential answers. Among the vehicles receiving the information, only those interested in the parking space answer to the coordinator. Each interested vehicle v_i provides to the coordinator its vehicle's identifier (see Section 2.1) and the information necessary for the coordinator to choose the vehicle to which the resource has to be allocated, such as the time t_i, corresponding to the current duration of the search of a parking space for that vehicle.

After a period of time T (amount of time during which the coordinator waits for answers from interested vehicles), the coordinator chooses, among the vehicles that answered, the one for which the parking space is "reserved". Different policies may be applied to choose the vehicle to which the parking space should be allocated. The choice of an appropriate allocation policy will be further discussed in Section 4 about experimental evaluation. Finally, the vehicle sends a message to the coordinator to acknowledge the reception of the coordinator's message and to confirm that it will take the parking space. This acknowledgement avoids losing parking spaces: in case the chosen vehicle does not accept the parking space, another vehicle is chosen by the coordinator. This exchange will be performed in a short time. However, if the acknowledgement should get lost, the parking space would be allocated to a second vehicle, generating some competition; this unlikely inconvenience is preferable to the possibility that the advertisement of the parking space gets lost.

In case no interested vehicle answered, a new coordinator is elected as discussed in the next section. The algorithms executed respectively by the coordinator and the vehicles searching for a parking space are detailed below:

```
    AT THE RECEIVER SIDE:
void declareInterest(Event apse, Coordinator c) {
/* Executed when a vehicle searching for a parking space receives an available-parking-space event apse from a coordinator c */
float ep = computeEP(apse);
if $ep \geq RelevanceThreshold$ then {
    send(c.identifier, getSearchDuration(), myID);
} }

    void confirmInterest(Event apse, Coordinator c) {
/* Executed when a vehicle is informed by a coordinator c that it has been allocated the parking space apse */
if (searching-for-a-parking-space) then {
    searching-for-a-parking-space = false;
    send(c.identifier, commitMessage, myID);
}
else
/* Parking space already found or the driver changed her/his mind */
    send(c.identifier, abortMessage, myID);
}

    AT THE COORDINATOR SIDE:
void advertiseParking(Event apse); {
/* Executed when a vehicle leaves a parking space or becomes a coordinator */
broadcast(apse, myID);
listen(T, answers);
if (answers $\neq$ null) then {
```

---

[2] If the event is generated by a fixed sensor in the parking space, an initial coordinator is chosen between the vehicles receiving this event.

```
    sort(answers); /* according to the allocation policy and the information communicated by the interested vehicles */
    i=0; notification=false;
    while (i < answers.size and notification==false) {
        notification = notify(answers[i++].identifier);
/* notification becomes true when the winning vehicle acknowledges the allocation */
    } }
else electNewCoordinator();} /* No interested vehicle so far */
```

3.2 Extending the Notification Range

We have described so far the interactions between the coordinator and the interested vehicles to allocate an available parking space. However, we have considered that at least one of the vehicles within the communication range of the coordinator was interested in finding an available parking space. If this is not the case, the information has to be relayed farther to try to find a vehicle interested in the resource. Anyway, it is not possible anymore to interact with the same coordinator using multi-hop techniques. Indeed, due to the use of short range communication networks and the absence of any fixed support infrastructure, we have no way to guarantee that the coordinator would be still reachable when using multi-hop relaying (all vehicles are highly mobile). Thus, the decision process could not be guaranteed.

Instead, we rather try to choose a new coordinator. The new coordinator has to be farther from the resource to increase the probability to reach new potentially interested vehicles. In the case of parking spaces, the coordinator should however not be selected too far away from the available slot. Indeed this would increase the probability that another vehicle arrives to park in that parking space in the meanwhile. Instead, we choose the new coordinator according to the demand in terms of available parking spaces in its vicinity.

Our goal is to find an interested vehicle as quickly and as close to the parking space as possible. Therefore, each vehicle periodically broadcasts to its neighbors its “state” (i.e., whether it is searching for a parking space or not). This allows each vehicle to estimate the approximate number of nearby vehicles searching for a parking space. So, when a new coordinator has to be elected, the former one broadcasts a message to its neighbors. The neighboring vehicles receiving that message (and not already coordinators for another parking space) reply to the coordinator by indicating their estimations. The coordinator then sorts the candidates in increasing number of these estimations and contact the vehicles in that list following that order until one vehicle confirms the reception of the proposal and so becomes the new coordinator. In case no candidate coordinator answers, the former one keeps its role and, after a while, broadcasts again the message about the available parking space. By then, new vehicles interested can now be in its neighborhood. If not, a process to switch the coordinator is initiated again. In the following, we summarize the algorithm executed to elect a new coordinator when needed.

```
AT THE COORDINATOR SIDE:
void electNewCoordinator() {
/* Executed when a coordinator wants to be replaced */
notification=false;
broadcast(coord, myID);
listen(T, answers);
if (answers ≠ null) then {
    sort(answers);
    i=0;
    while (i < answers.size and notification==false) {
        notification = notify(answers[i++].identifier); /* notification becomes true when a vehicle accepts to become the new coordinator */
    } }
if (notification == false) then {
    wait(T);
    advertiseParking(apse);
} }
```

Thus, vehicles interested in finding an available parking space can be located even if they are further from the available parking space than the communication range of the wireless network used. Each vehicle receiving a notification about an available parking space has to verify if that information is relevant. Therefore, the vehicle computes the Encounter Probability presented in Section 2.2 for the event received, to estimate if it can reach the space while it is still estimated available. In that case, it declares its interest in the parking space.

3.3 Some Final Remarks

Obviously, several available parking spaces can be communicated to the same vehicle at the same moment (e.g., two close vehicles can leave their parking space or become coordinators at the same time). Thus, a vehicle can receive different messages, issued by different coordinators, indicating an available parking space. Then, a difficulty arises for that vehicle to choose to which coordinator it should send a positive answer. Indeed, the vehicle does not know by which coordinator it could be elected since it does not even know which other vehicles will answer. Consequently, we choose to let the vehicle answer to all the available coordinators. Once again, the acknowledgement presented previously is used to avoid losing a parking space in case the same vehicle is elected twice by two different coordinators. At the coordinator side, if one vehicle does not confirm its interest in the parking space, the coordinator tries to contact the second vehicle in the list of answers, and so on.

It should also be noted that a vehicle that has been allocated a parking space could find and take a different parking space before reaching it. In that case, it can advertise again its allocated parking space, acting as a coordinator for it. In this way, the information about the availability of the parking space is not lost.

4. EXPERIMENTAL EVALUATION

We evaluated different strategies to advertise available parking spaces on a parking lot. Our goal was both to evaluate the efficiency of our EP mechanism to determine the relevance of available parking spaces and to observe the impact of our reservation protocol in this context.

We have developed a prototype of VESPA[3] and performed some experiments in a real scenario (see Figure 2). However, due to obvious scalability reasons (it is difficult to perform repeatable scenarios with a high number of vehicles in a real environment), we use a simulator that we have developed to evaluate our system. Tests in a real environment are thus used mostly for verification and to calibrate our simulations.

---

[3] see `http://www.univ-valenciennes.fr/ROI/SID/tdelot/vespa/prototype.html`

Figure 2: Testing VESPA in a real environment

4.1 Simulator description

In order to evaluate our solution with an important number of vehicles, a simulator has been designed. We needed a testing system that could simulate realistic vehicles' movements, wireless exchanges, generation of events, etc. Moreover, whereas the traffic naturally strongly impacts the dissemination of resources among vehicles, the information exchanged among those vehicles also influences the way the vehicles move. For example, if an interested driver receives the position of an available parking space, s/he will move in that direction. This is why we decided to develop our own simulator, since no existing one could fit our needs.

Our simulator has been developed not only for scenarios with parking lots but to simulate any scenario that can be interesting to evaluate a VANET application. The vehicles drive from a random departure point to a destination point through roads defined according to real maps. The choice of roads used for each vehicle to reach its destination is computed using Dijkstra's shortest path algorithm. When a vehicle does not have any destination point because it is looking for a parking space but does not know any available slot, a random road is chosen (simulating a driver that is looking around for a place to park).

It should be emphasized that the behavior of the vehicles during the simulations depends on both the information provided by the inter-vehicle communication system (i.e., the positions of available parking spaces communicated to the driver) and the information that the drivers observe. For example, to simulate a real environment, a driver searching for a parking space should take an available one once s/he sees it, even if a farthest one has been allocated to her/him.

4.2 Experimental Results

In the following, we present some of the results obtained thanks to our simulations. Obviously, relevance functions other than the Encounter Probability could be used. For instance, in [11], the authors use the following relevance function F to characterize the relevance of a parking space s:

$$F(s) = -\alpha \times t - \beta \times d \qquad (\alpha, \beta \geq 0)$$

where t is the age of s, d is the distance from the location of s, and α and β are non-negative constants that represent the relative importance of time and distance.

We compare different strategies for relevance evaluation during our simulations:

- One with no inter-vehicle communication (i.e. drivers searching only park their car when they "see" an available space), called *View Only*.
- One for which the relevance is evaluated using the relevance function proposed in [11], called *Time and Distance*.
- One for which the relevance is evaluated using our Encounter Probability, called *VESPA only*.

The three previous strategies, considered alone, imply that no reservation protocol for parking spaces is used. Besides, different allocation strategies are considered:

- An available space is allocated by the coordinator to the vehicle with the highest Encounter Probability for the considered space, called *Reservation EP*.
- The relevance function F(s) used by the *Time and Distance* strategy is considered to allocate available spaces to vehicles, called *Reservation Time and Distance*. Our goal is here to highlight that our reservation protocol can be used on top of systems other than VESPA.

During our experiments, we observed for each solution:

- the average time needed to find a parking space;
- the variance of the time needed to find a parking space;
- the number of resources displayed to drivers;
- the percentage of useful information displayed;
- the number of vehicles which found a parking space thanks to the system (i.e. not because their drivers saw by themselves an available parking space).

.

In the following, we present our results with a scenario of a parking lot with 60 parking spaces. Each vehicle entering the parking lot was considered as searching for an available space. During our experiments, some vehicles leave their parking spaces while others (driving at 10 kmph) are searching for an available slot. We evaluated different configurations with more or less free parking spaces and searching vehicles. For the configurations where the number of available parking spaces is always greater than the number of vehicles searching, the strategies using V2V communications (i.e., all those we evaluated except *View Only*) perform better but no significant difference between them can be observed. So, we present in the following the results for a "heavy" parking configuration where we always consider more searching vehicles than available parking spaces. This configuration is one of the most interesting and challenging ones because resources are rare. It is so particularly important to allocate them in the best way.

In this "heavy" configuration, we considered a number of vehicles searching for a parking space ranging between 1 and 10% of the total capacity of the parking. The penalty coefficients used to compute our Encounter Probability are: $\alpha^{-1} = 50$, $\beta^{-1} = 30000$, $\gamma^{-1} = 600$ and $\zeta = 0$ in order to prefer free parking spaces located in the same row, even if

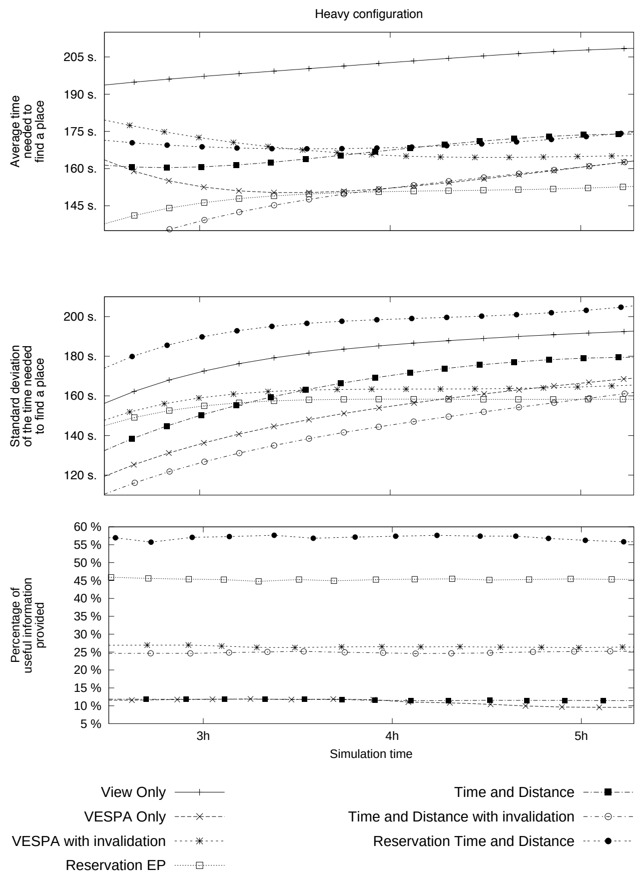

Figure 3: Experimental results

there are closer ones (considering the Euclidean distance) in the neighboring ones.

We used $\alpha^{-1} = 30000$ and $\beta^{-1} = 50$ for the coefficients of the function $F(R)$ used by the *Time and Distance* strategy, in order to have the same relative weight. Indeed, since there are always several vehicles searching for a parking space in our configuration, the penalty should consider the distance between the vehicle searching and the parking space rather than the age of the resource. As concerns the network aspects, the waiting time D used by the reservation protocol is set to 500 ms, the propagation time (i.e., time needed for a packet to travel from the sender to the receiver) is set to 50 ms and the communication range is 200 m.

Figure 3 presents the average results obtained after 20 simulations for three evaluated criteria: the evolution of the average time needed by the vehicles to park, the standard deviation of the amount of time needed to find a parking space, and finally the percentage of useful information provided to the driver (i.e., parking spaces allocated to a driver that are really obtained by that driver).

As concerns the results of the strategies without reservation protocol, we can first observe, thanks to the average times, the interest of V2V-based solutions. Besides, the results for the *Distance and time only* approach are not very good in terms of average search time, as that relevance function was basically proposed to monitor a set of available parking spaces close to the vehicle. Compared with *VESPA only*, we can notice the importance of the vehicles' direction to determine the best parking space to allocate, since this may avoid u-turns. Moreover, it also maximizes the probability that a vehicle arrives to a parking space earlier than other vehicles trying to park. So, an available parking space in the same row should be preferred even if it is not the closest (i.e., one available parking space may be closer in the next row but the time to reach it will be higher).

To provide vehicles more relevant information, we introduced invalidation messages. These messages are generated by vehicles, once parked, in order to inform the other ones that the slot they chose is not available anymore. The results for the *Time and Distance with invalidation* and *VESPA with invalidation* strategy are presented in Figure 3. These invalidation messages really improve the result for the *Time and Distance* strategy, since they avoid wasting time to reach a no more free parking space. However, these messages also have an impact on the network load. They indeed require the generation of one additional message per parking space. Furthermore, this message has to be broadcast using multi-hop techniques in order to (try to) reach all the vehicles previously informed of the available parking space.

Regarding the reservation protocol, the average times are rather good, compared with the other solutions, specially when the Encounter Probability is used to allocate parking spaces. Indeed, since we considered that a driver who sees a parking space will park on that space even if s/he had another parking space allocated, it is better to allocate a parking space close to the vehicle (and if possible in its driving direction). Otherwise, the probability that the parking space is going to be occupied by another driver increases. In that case, the vehicle which had the parking space allocated would have to ask for a new allocation when it receives a new event about an available parking space. This explains the average performance of the reservation protocol deployed on top of the *Time and Distance* strategy due to the "row effect". Indeed, the target parking space is here computed with the Euclidean distance and may be located in the next row. The driver may so need more time to reach it what increases the probability the space is no more available when s/he arrives. This also justifies the important standard deviation for this strategy. Besides, the reservation protocol allows maximizing the percentage of useful information provided (i.e., the ratio between the number of resources effectively acquired among the total number of resources communicated to the driver), whatever the allocation strategy used. This means that the reservation protocol allocates a lot of useful parking spaces (i.e., where the driver receiving the position of the available parking space can effectively park her/his vehicle) even if the misses may be very penalizing using the *Time and Distance* allocation strategy (i.e., the driver needs a lot of time to note that the parking space is no more free). Notice that this percentage of useful information is unsurprisingly bad for the strategies where no reservation protocol is used. This means that the drivers received more non-relevant information because the positions of free parking spaces are not communicated to only one driver but to several ones at the same time.

We can conclude that the *Reservation EP* strategy is the best choice with the heavy parking lot configuration. This strategy helps to reduce the average search time for a parking space, especially when the competition is high. By reducing that competition, it also improves the percentage of information communicated to the vehicles is actually useful. Indeed, in spite of the congested situation (in terms of vehicles searching for an available parking space), about 50% of the vehicles receive a correct information as opposed to the strategies without reservation protocol (about 15%). We have performed other experiments with different number of vehicles and parking spaces, leading to similar conclusions.

5. RELATED WORKS

Some works focus on disseminating parking space information in a VANET using inter-vehicle communications, such as [11, 3] or the SmartPark project (see `http://smartpark.epfl.ch/`). One of the key issues in these works is defining an appropriate relevance function to determine the way the data should be disseminated. Moreover, [11] emphasizes the importance of considering aggregate information and guiding the driver towards areas where finding an available parking space is very likely (instead of guiding the user towards a specific parking space that could be taken by another vehicle in the meanwhile). However, none of these works actually allocates parking spaces to vehicles and so competition issues may arise (e.g., with many vehicles going to park to the same area the number of available parking spaces will drop quickly). Related to the concept of "relevance", [2] proposes a mechanism to estimate the probability that at least one parking space in a given parking lot will be available when the driver arrives there. Instead of disseminating data to potential interested vehicles, [10] proposes a system where the vehicle asks an *RSU* (*Road Side Unit*) about the availability of parking spaces. This work studies possible security vulnerabilities and relies on a support infrastructure of RSUs. A query-based approach is also presented in [1], which groups parking meters (equipped with a low-cost processor) in clusters. These parking meters can answer queries from vehicles passing by, but a vehicle must be within range of some parking meter to receive this information. Moreover, no experimental evaluation is presented. In all the works presented so far, the dissemination of information could lead to competition for the parking space.

Some works also consider the importance of booking a parking space for a vehicle, to avoid the competition problem. For example, in [8] an agent-based parking reservation facility is presented. However, it is adapted to the specific scenario of a campus and relies on the existence of a support infrastructure (composed of *InfoStations*) and a centralized computer (an *InfoStation Center*) where the information about the parking spaces is managed.

To conclude this section, we will mention some other interesting works less related to our approach. For example, [5] focuses on a scenario where parking prices can be negotiated and proposes an multi-agent system to perform the negotiation and select an optimal parking space for a vehicle. However, a global communication infrastructure is assumed. Finally, [6] presents an agent-based model to study the interrelation of different factors, such as the parking pricing strategies and the behavior of drivers.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a solution to disseminate and allocate available parking spaces to drivers. The originality of our contribution resides in our reservation protocol, which is to the best of our knowledge the first solution to allocate parking spaces to vehicles in VANETs. We have evaluated our approach in both real and simulated scenarios, obtaining positive results: our system reduces both the time needed to find a parking space and the competition among the vehicles.

The reservation protocol on top of the *Distance and time* relevance function [11] does not provide good results with that parking lot configuration. However, such a strategy could be efficient in urban areas to allocate available parking spaces located on small streets around the main boulevards. Since the traffic is less important on such streets, the driving direction would not be so penalizing. However, in any case, it would be interesting to introduce the search time of the vehicles in the resource allocation process (in addition to the relevance function), in order to give priority to the vehicles searching for the longest time. We are now extracting streets of the city of Valenciennes from the Postgis/TeleAtlas database connected to our simulator in order to evaluate our reservation protocol in other scenarios.

7. REFERENCES

[1] P. Basu and T. Little. Wireless ad hoc discovery of parking meters. In *MobiSys Workshop on Applications of Mobile Embedded Systems (WAMES'04)*, June 2004.

[2] M. Caliskan, A. Barthels, B. Scheuermann, and M. Mauve. Predicting parking lot occupancy in vehicular ad hoc networks. In *Vehicular Technology Conference (VTC'07)*, pages 277–281. IEEE, 2007.

[3] M. Caliskan, D. Graupner, and M. Mauve. Decentralized discovery of free parking places. In *Third ACM International Workshop on Vehicular Ad Hoc Networks (VANET'06)*, pages 30–39. ACM, 2006.

[4] N. Cenerario, T. Delot, and S. Ilarri. Dissemination of information in inter-vehicle ad hoc networks. In *2008 IEEE Intelligent Vehicles Symposium (IV'08)*, pages 763–768. IEEE, June 2008.

[5] S.-Y. Chou, S.-W. Lin, and C.-C. Li. Dynamic parking negotiation and guidance using an agent-based platform. *Expert Systems & Applications*, 35(3):805–817, 2008.

[6] M. Dell'Orco and D. Teodorović. *Applied Research in Uncertainty Modeling and Analysis*, volume 20, chapter Multi Agent Systems Approach to Parking Facilities Management, pages 321–339. Springer, 2005.

[7] T. Delot, N. Cenerario, and S. Ilarri. Estimating the relevance of information in inter-vehicle ad hoc networks. In *2008 MDM International Workshop on Sensor Network Technologies for Information Explosion Era (SeNTIE'08)*, pages 151–158. IEEE , April 2008.

[8] I. Ganchev, M. O'Droma, and D. Meere. Intelligent car parking locator service. *Int. Journal "Information Technologies and Knowledge" (IJ ITK)*, 2(2):166–173, 2008.

[9] J. Luo and J.-P. Hubaux. A survey of research in inter-vehicle communications. In *Embedded Security in Cars - Securing Current and Future Automotive IT Applications*. Springer, 2005.

[10] R. Panayappan, J. Trivedi, A. Studer, and A. Perrig. Vanet-based approach for parking space availability. In *Fourth ACM International Workshop on Vehicular Ad Hoc Networks (VANET'07)*, pages 75–76. ACM, 2007.

[11] B. Xu, A. M. Ouksel, and O. Wolfson. Opportunistic resource exchange in inter-vehicle ad-hoc networks. In *Fifth International Conference on Mobile Data Management (MDM'04)*, pages 4–12. IEEE, 2004.