ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2016, T. 45, Nr. 2

Proactive Discovery and Management of Ride-Sharing Opportunities in Smart Vehicular Ad-Hoc Networks

Jack F. Bravo-Torres

Área de Ciencias Exactas, Universidad Politécnica Salesiana
Calle Vieja 12-30 y Elia Liut, Cuenca, Ecuador
e-mail: jbravo@ups.edu.ec

Yolanda Blanco-Fernández, Martín López-Nores, José J. Pazos-Arias, Manuel Ramos-Cabrer, Alberto Gil-Solla

AtlantTIC Research Center for Information and Communication Technologies
Departamento de Enxeñaría Telemática, Universidade de Vigo
EE Telecomunicación, Campus Universitario s/n, 36310, Vigo, Spain
e-mail: yolanda@det.uvigo.es, mlnores@det.uvigo.es, jose@det.uvigo.es,
mramos@det.uvigo.es, agil@det.uvigo.es

crossref http://dx.doi.org/10.5755/j01.itc.45.2.8464

Abstract. Ride-sharing promotes a way to better use empty seats in vehicles, thus saving expenses and reducing emissions of greenhouse gases. In most of the current ride-sharing portals over the Internet the users must explicitly enter information about origin, destination, route, time and date when searching for riders who fulfill their mobility needs. In this paper, we explore new opportunities of ride-sharing to proactively discover the most frequent trips of each user and automatically selecting trip mates for each itinerary. To this aim, we exploit the large number of people who gather together in heavily trafficked zones at certain times to deploy a smart Vehicular Ad-Hoc Network (VANET) over their handheld devices. The smart VANET exchanges among the vehicles the information necessary for (i) matching the users' itineraries and particular preferences, and (ii) identifying like-minded riders for common routes. The approach has been validated by a VANET simulator and a prototype that was used by 46 users.

Keywords: Ride-sharing; vehicular ad-hoc networks; context-awareness.

1. Introduction

Increasing costs and persistent traffic problems make it necessary to develop affordable and flexible solutions for managing the mobility of people. Currently, many government bodies of diverse countries promote ride-sharing initiatives (aka carpooling, liftsharing or covoiturage) to fight problems of their surface transport networks. Specifically, several countries charge reduced toll fees to vehicles with more than 3 or 4 occupants (the so-called HOVs, *High Occupation Vehicles*), while others provide special lanes on the roads or reserved spaces in parking lots. These measures have great potential to improve the capacity of congested traffic corridors with minimal investment in infrastructure, thus cutting down energy consumption and reducing emissions of greenhouse gases. Some examples of ride-sharing portals currently available over the Internet are *www.blablacar.es* and *www.vao.es* (Spain), *www.carpool.com.pt* (Portugal), *www.zimride.com* (Canada, USA and Mexico), *www.rideshare.co.uk* (UK), *www.mitfahrgelegenheit.de* (Germany), and *www.123envoiture.com* (France).

Likewise, ride-sharing has gained momentum in the scientific community during the last years [8, 12, 28], where the researchers have proposed initiatives that require (i) the drivers to *explicitly* enter information about the trips they will be making (basically, origin, destination, route, time and date), and (ii) the potential passengers to *make searches*, expressing their mobility needs along with spatio-temporal proximity criteria. After discovering one possibility to make a shared ride, the two parts get in contact to agree on the details of the trip: exact meeting point and destination, amount of money that the passenger will give to the driver, space needed for luggage, etc. When the date and time come, driver and passenger meet and share the former's vehicle as agreed.

In this paper we explore new opportunities of ride-sharing aimed at *proactively* discovering trips the users make frequently and automatically selecting riders for those routes. To this aim, our approach establishes sporadic networks among people who are physically close to each another at a certain moment, by supporting ad-hoc connections among their mobile devices. To put it from another angle, we deploy a smart Vehicular Ad-Hoc Network (VANET) over nearby cars on the road to exchange among the vehicles the information necessary for (i) matching the users' itineraries and particular preferences, and (ii) identifying like-minded riders potentially interested in sharing car along common routes. To this aim, we exploit the fact that there exist heavy traffic zones that bring together a huge number of people every day at certain times (e.g. when they start/finish working days at peak hours, leave/pick up children at school, or attend leisure activities during weekends). This situation turns VANETs into propitious environments to proactively identify ride-sharing opportunities.

Provision of these functionalities requires mechanisms to ensure that the VANET is a reliable communication environment, where the cornerstone are the multihop ad-hoc connections established among the users' mobile terminals. Besides, bearing in mind the computational restrictions of these handheld terminals, our approach also needs schemes aimed at allocating among the VANET's nodes (vehicles) the tasks and computations required in the organization of shared rides. Thanks to these refinements, our research contributes with the development of complex information services that remained unexplored up to now in the realm of ad-hoc networks *without any pre-established infrastructure*. In this regard, most of the applications deployed in a vehicular environment relies on the availability of an Internet infrastructure that is readily accessible via technologies such as WiFi, WiMAX and 3G, among others [13]. However, in this paper we deal with vehicular ad-hoc networks without infrastructure to allow the users to enjoy enhanced services even when their mobile devices have no permanent Internet connection. This way, we exploit the fact that the vehicles are communication nodes that can continue to operate autonomously when it is not possible to connect to the Internet.

The paper is organized as follows. In Section 2 we review related works in the scope of ad-hoc networks and their applications. Section 3 presents the architecture of the ride-sharing platform proposed, which hosts security schemes in order to ensure proper identification of riders to each other, and compensation policies in case of agreement violations. In Section 4 we focus on the refinements of our smart VANET, by describing the matching algorithms that (i) automatically identify often-traversed routes for each user, and (ii) select his/her most appropriate trip mates. After illustrating an example scenario and detailing our experimental evaluation in Section 5 and Section 6, respectively, we highlight the main conclusions and further work in Section 7.

2. Related Works

One of the most commonly adopted Mobile AdHoc Networks (MANETs) are the wireless sensor networks (WSNs), which consist of spatially distributed autonomous sensors that act as network nodes that collect and send environmental data to a main location [6, 27]. With the development of new communications and mobile technologies, many local governments have decided to exploit the presence of WSNs in their cities to efficiently manage several applications in their daily responsibilities. The enormous amount of information collected by sensor devices allows the automation of several real-time services to improve city management by using intelligent traffic-light patterns during rush hour, reducing water consumption in parks, or efficiently routing garbage collection trucks throughout the city [7, 10].

WSNs have also been intensively adopted in military communication applications, due to the fact that MANETs make it possible to set up communication environments in inaccesible terrains placed in remote war zones [1, 16]. Likewise, the proliferation of WSN technologies has turned context-awareness into a must in many non-military applications, ranging from approaches to helping tourists to find interest points in a city and to navigate through them by mobile phones and NFC technologies [5], to solutions that combine 3D visualization techniques and Zigbee sensor devices to build context-aware applications in smart environments [17]. *Smart city* technologies have also provided substantial benefits to improve people's daily lives. As an example in this field, some authors have investigated how exploit inductive magnetic sensing and Wi-Fi scanning for traffic planning purposes [18].

Regarding the vehicular environment, this field has attracted researchers from diverse areas to develop applications, protocols and simulation tools for VANETs [2]. VANETs are classified as an application of MANETs, being decentralised, self-configuring networks of mobile nodes that exchange information without relying on any pre-established infrastructure. In a vehicular environment, Hartenstein and Laberteaux wrote a survey on aspects related to communication and networking, as well as security and privacy issues [14]. Meanwhile, many researchers investigated routing protocols for VANETs and their requirements to achieve better communication time with less consumption of network bandwidth [19,20]. Regarding VANET applications, most of the existing approaches exploit wireless communication between vehicles (or between vehicles and infrastructure) to improve road safety and avoid accidents [2, 13]. Many of these applications disseminate (through flooding) and gather information about the vehicles on the road, thus providing real-time traffic data to drivers [11, 24]. However, there are also some entertainment applications aimed to improve drivers and passengers comfort levels and enhance traffic efficiency, providing information about weather or traffic conditions [15] and about the location of the nearest hotel or petrol station [33], and even enabling to discover and book free cabs through vehicle multihopping [35]. Some of these emerging VANET applications rely on roadside access points (e.g., at petrol stations or restaurants) to provide content downloading services. In this scenario, the mobile users are encouraged to cooperatively assemble the files through a P2P file sharing, so that some vehicles access the content through the Internet and later distribute them to the rest of cars over the VANET [25, 26]. Although these applications enable the deployment of a wide variety of advanced services to vehicles, their potential has not been totally exploited yet in scenarios *without permanent Internet access* [23, 29]. Our approach faces this situation by contributing with advanced information services based on establishing sporadic networks among nearby users on the road, which are completely novel in the scope of the vehicular ad-hoc networks *without any infrastructure*.

3. The Platform Architecture

The architecture of our ride-sharing platform is depicted in Figure 1. Here we can see that the cornerstone is a smart VANET that is deployed over the handheld devices of the users on the road. Its potential is exploited to identify nearby users with similar preferences whose mobility needs match against a given user's often-traversed routes. The final terms of the shared trip (e.g. meeting point, price, time...) can be discussed through a PBX system (whose technical internals fall out of the scope of this paper) that allows the riders to get in touch with one another. This information is recorded by a manager system that keeps one account for each user to store personal data, valid bank account or credit card data (in order to charge/pay for the realisation of shared rides), along with contact information (such as e-mail addresses).

The manager system stores this information in order to track the trips in which the user has participated or will participate, and to maintain his/her credit balance. Credit is initialised with a certain amount of money upon registering, and can be updated at any time by transactions that involve the financial institutions that manage the user's money and payment gateways, as depicted in Figure 1.

The architecture of the smart VANET has been designed with an eye towards the capability of *reusing the improvements we have developed* for a vehicular environment without any pre-established infrastructure *out of the scope of the ride-sharing initiatives*. To this aim, as shown in Figure 1, we propose a multilayer architecture where the specific features of each domain are placed in an application layer, whereas the lower layers support the modeling and management of the knowledge about the users as well as the algorithms for routing traffic over the ad-hoc network. Specifically, we have defined the following layers in our smart VANET:

- **Application layer:** this level provides the riders with the interfaces necessary for entering information about their personal interests and mobility preferences, and accessing the security mechanisms that ensure their proper identification and compensation in case of service violation.
- **Knowledge management layer:** this level supports (i) the modeling of the users' profiles that are locally stored in their handheld devices, and (ii) the context-aware algorithms adopted for matching the potential riders' mobility needs as per the itineraries of their respective trips and even their personal preferences.
- **Distribution layer**: since we deal with a VANET without any pre-established infrastructure, the matching algorithms hosted in the knowledge management layer must run on the users' handheld devices, ensuring, at the same time, computational feasibility and scalability. For that reason, the distribution layer needs to provide mechanisms to manage and share the resources available in each mobile device with the goal of effectively distributing communications and computations over the ad-hoc network. To this aim, the distribution layer manages multiple instances of the same matching algorithm that work with different profiles. Specifically, the instance running in a user's device matches his/her local profile against other riders' profiles that are received through the ad-hoc network.
- **Ad-hoc communications layer:** this level routes the traffic generated in the upper levels over the nodes (vehicles) of our smart VANET. For that purpose, the ad-hoc communication level relies on a virtualization layer named VaNetLayer that works in tandem with the routing protocol VNAODV+, which have been specifically developed by considering the particular requirements of a VANET. We will just take a quick look about the functioning of the VaNetLayer and VNAODV+ in the next section, before focusing on the remaining layers of the smart VANET (ranging from the distribution layer to the application one).

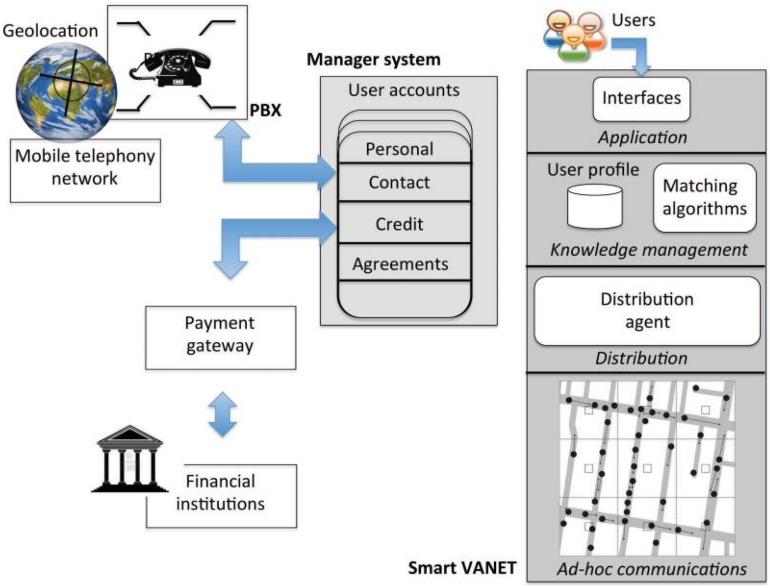


Figure 1. Main elements of our ride-sharing platform over a smart VANET

4. Proactive Discovery of Shared Rides in VANETs

4.1. Ad-hoc Communication layer

VaNetLayer and VNAODV+ are refinements we have developed starting from the VNLayer by Dolev *et al.* [9] and from a virtualized version of the routing protocol AODV (named VNAODV) by Wu [34], which were proposed to handle communication in generic MANETs. Our experiments by the VANET simulator presented in [4] revealed several sources of inefficiency both in the VNLayer constructs and in VNAODV. These are features that did not cause significant trouble in MANET simulations but did have an impact when faced with some peculiarities of VANETs, namely the comparatively faster movements of the nodes and the fact that these nodes are not subject to the strict energy, space and computing capabilities restrictions of MANETs.

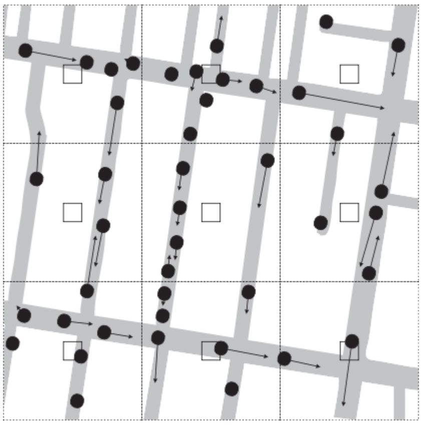


Figure 2. Static virtual nodes (white squares) overlaying the moving vehicles of a VANET (black circles)

Next, we sketch the operation of our VaNetLayer and the routing protocol VNAODV+. Algorithmic internals about the refinements we have included to improve their respective performance in a vehicular communication environment can be found in [3].

The VaNetLayer is based on engaging the mobile *physical nodes* (PNs) in collaboration to emulate *virtual nodes* (VNs) that remain in known fixed locations. As seen in Figure 2, the VaNetLayer divides the geographical area of an ad-hoc network into square regions, whose size is chosen so that every PN (denoted by a black circle) in a region can reliably send and receive data from every other physical node in that region and neighboring ones. The VNs (represented as white squares) can be thought of as lying in the center of the corresponding regions (one VN per region), being able to send messages directly to up to 8 neighbouring VNs, as depicted in Figure 2. Each VN is emulated by the PNs located in the corresponding region, using distributed algorithms based on replicated state machines.

One PN is chosen as the *leader* in the region and becomes the primary responsible for packet reception, buffering and forwarding. Meanwhile, a subset of non-leader nodes are designated as *backups* to maintain replicas of the relevant state information from the upper layers, consistent with the leader's version. This way, the VNs can maintain persistent state and be fault tolerant even when individual physical nodes fail or leave the region.

Regarding the routing process, AODV is one of the most widely adopted reactive protocols in ad-hoc environments, meaning that routes are created only when there is a request. Briefly, a Request message (RREQ) is broadcast through intermediate nodes, so that the flooding goes on until reaching the destination node or a node that knows a valid route to it. That node generates a Reply Request message

(RREP), which is sent to the source node by unicast through the backpath just learnt. The main drawbacks of AODV as a MANET routing protocol have to do with (i) the overhead caused by broadcasting RREQ messsages, and (ii) the unstability of the routes in scenarios of high mobility. We tackle both problems in VNAODV+ with the abstraction of the VaNet-Layer, by covering the network area with static virtual nodes that act on behalf of a comparatively greater number of physical ones. Our VNAODV+ routing algorithm retains the core of AODV, including the packet types and the settings on most parameters and incorporating the necessary changes to have data sent through VNs rather than PNs.

As argued in [3], the refinements of our VaNet-Layer and the routing algorithm VNAODV+ support the traffic generated in the upper layers, which mainly includes the users' profiles and the control information (overhead) required for their distribution over the vehicular ad-hoc network, as we will describe later. Likewise, our mechanisms allow TCP to work efficiently at transport level, providing reliable delivery of packets and managing the messages and timers involved in the communication of the applications.

4.2. Distribution layer

As depicted in Figure 1, the cornerstone of the distribution layer is the so-called *distribution agent* that manages both (i) the proactive requests aimed to find trip mates for a particular user and regular route, and (ii) the responses received over the VANET. The goal is to balance the traffic handled by the distribution layer in order to ensure the operating of the smart VANET established among the users' mobile terminals. For that purpose, we have defined a procedure involving multiple messages referred to requests, responses and acknowledgements, as well as timers.

Specifically, the following procedure is triggered after discovering a regular route for a given user (which we will refer to as *target user*):

1. First, the distribution agent running on the target user's terminal (denoted hereafter by *target agent*) composes a RIDER_REQUEST message where includes the local profile with his/her mobility needs and personal preferences. Also, a TTL (Time To Live) field is included in the package header which limits its lifetime over the ad-hoc network, thus preventing this message from circulating indefinitely over it. As usual, this field has been implemented as a counter that is decremented in each hop, so that the request is discarded when its value is 0. The RIDER_REQUEST message is broadcast over the VANET and a timer is started to define how long the target agent will wait for possible responses (denoted by WAITING_RIDER_RESPONSES). Although the VANET is disconnected and the target agent does not have any neighbors to forward or respond to the RIDER_REQUEST message, the request does not need to be broadcast more than once over the ad-hoc network. This is thanks to the mechanisms deployed in the VaNetLayer, which are responsible of message retransmission (when the VANET is active/connected again) and even of avoiding possible data losses by working in tandem with TCP protocol. The details can be found in [3].

The aforementioned request process is exemplified at the top of Figure 3, where we assume TU as target user and R_1, R_2, R_3 and R_4 as candidate riders for a shared trip.

2. The broadcast RIDER_REQUEST message reaches all the nodes (terminals) of the VANET. The distribution agent running on each candidate rider's terminal (hereafter *candidate agent*) compares his/her personal profile against the one received from the target agent. For instance, in Figure 3, a matching level must be computed between the preferences of R_1 and TU in R_1's terminal, between R_2's preferences and TU 's profile in R_2's terminal, and so on. To this aim, we adopt the matching algorithms placed in the knowledge management layer in Figure 1, which will be described in Section 4.3. In case of compatible mobility needs and preferences, the candidate agent sends via unicast a RIDER_RESPONSE message to the target agent, including an Id that enables the candidate rider to be uniquely identified within the platform. In the example shown in Figure 3, the messages 2, 3 and 4 received by TU 's agent reveal that only the itineraries and preferences of R_1, R_2 and R_3 match against TU 's ones.

3. In case of receiving multiple replies, the target agent needs to check whether his/her possible trip mates are also compatible with each other, by means of a group matching process that considers their respective personal preferences[1]. Again, this group matching process is distributed over the terminals in the VANET, so that each user's terminal matches his/her local profile against other rider's profile received over the ad-hoc network, similarly to what we explained before. The target agent is in charge of managing and organizing this process. To this aim, the target agent notifies to each rider's terminal the unique Id of the user whose profile must request and compare against the one locally stored. For instance, in the scenario illustrated in Figure 3, the group matching must compare the preferences of (i) R_1 and R_2, (ii) R_2 and R_3, and (iii) R_1 and R_3, respectively. For that purpose, TU 's agent organizes the group matching so that (i) R_1's terminal compares the profiles of R_1 and R_2, (ii) R_2's terminal matches the profiles of R_2 and R_3, and (iii) R_3's terminal compares the profiles of R_3 and R_1.

4. Actually, TU 'agent notifies the Id to each possible trip mate by sending via unicast a

[1] Taking into account the capacity of an average vehicle, the group matching process would involve a reduced number of nodes (typically lower than or equal to 5) where, obviously, the role taken by each user (to ensure, for instance, a driver) and the number of available seats are also considered.

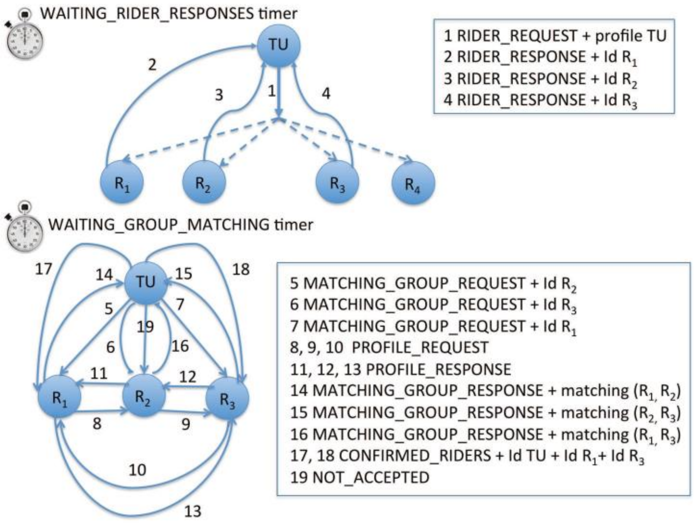

WAITING_RIDER_RESPONSES timer
TU
R1
R2
R3
R4
1
2
3
4
1 RIDER_REQUEST + profile TU
2 RIDER_RESPONSE + Id R1
3 RIDER_RESPONSE + Id R2
4 RIDER_RESPONSE + Id R3
WAITING_GROUP_MATCHING timer
5 MATCHING_GROUP_REQUEST + Id R2
6 MATCHING_GROUP_REQUEST + Id R3
7 MATCHING_GROUP_REQUEST + Id R1
8, 9, 10 PROFILE_REQUEST
11, 12, 13 PROFILE_RESPONSE
14 MATCHING_GROUP_RESPONSE + matching (R1, R2)
15 MATCHING_GROUP_RESPONSE + matching (R2, R3)
16 MATCHING_GROUP_RESPONSE + matching (R1, R3)
17, 18 CONFIRMED_RIDERS + Id TU + Id R1+ Id R3
19 NOT_ACCEPTED

Figure 3. An example of how computations and communications are distributed over the ad-hoc network as per the guidelines driven by the distribution agents

MATCHING_GROUP_REQUEST message, as shown in messages 5, 6 and 7 of Figure 3. Also, a timer is started to define how long the target agent waits for the group matching results (WAITING_GROUP_MATCHING).

5. After receiving the MATCHING_GROUP_REQUEST from the target user, each candidate agent requests the corresponding profile by sending a PROFILE_REQUEST package, as depicted in messages 8, 9 and 10 in Figure 3.

6. Once the requested profile is sent (via a PROFILE_RESPONSE message), the matching process is carried out. As shown in Figure 3, R_2's candidate agent sends his/her profile to R_1's agent to compare the preferences of both riders (message 11). Something similar happens with the candidate agents running in the terminals of R_3 and R_1 (messages 12 and 13, respectively).

7. Finally, the result of each matching process is notified from each candidate agent to the target agent via a MATCHING_GROUP_RESPONSE message (messages 14, 15 and 16 in Figure 3). This way, once the WAITING_GROUP_MATCHING timer has expired, the target user knows the definitive list of compatible trip mates. This information is notified to each rider by sending via unicast either a CONFIRMED_RIDER message (including the Ids of all the trip mates) or a NOT_ACCEPTED message (in case of incompatibility). For example, in view of messages 17, 18 and 19 in Figure 3, the preferences of riders R_2 and R_3 are not compatible each other, which is notified to R_3 via the NOT_ACCEPTED message. The CONFIRMED_RIDERS messages received by R_1 and R_3 reveal that these users will be TU 's trip mates. The final terms of the trip will be agreed via the PBX.

4.3. Knowledge Management layer

Just as its name suggests, this layer manages the knowledge available about the potential riders, including the profiles to model their preferences and the algorithms to match their respective itineraries and particular interests. Actually, our approach checks firstly whether there exists overlapping among the routes of the considered users; if so, their preferences are compared to ensure a pleasant trip among like-minded riders. After presenting the user profiles, both matching algorithms will be described in Section 4.3.2 and Section 4.3.3, respectively.

4.3.1. The Users' Profiles

Information about the users of our ride-sharing platform are locally recorded in their personal profiles, which are organized as follows:

- *Personal information*: this part of the profile keeps the credentials of the user, including, for instance, his/her login, password and unique Id.
- *Trips and shared rides*: our profiles record information about both the trips made by each user (from which his/her commonly traversed routes can be inferred by exploiting the GPS-style navigation equipment of his/her mobile device) and the shared rides that our platform has proactively organized for him/her. Specific information for each shared ride is stored, including the personal Ids of the user's trip mates, details about the routes (e.g. origin, destination, meeting point) and scheduling information (time and date departure).
- *Mobility preferences*: during the registration, the user can define certain preferred features regarding the shared rides (such as shortest distance or time,

type of road, kind of vehicle, toll avoidance, extra distances that drivers would be willing to tolerate in picking up and dropping off riders, tolerances regarding distances and times...) and also about his/her potential trip mates (e.g. smoking/non-smoking carpooling partners, gender, age range...).

- *Personal interests*: the user can also indicate their particular interests by rating a set of preestablished categories (e.g. sports, politics, technology, music...). Our matching algorithms allow to adjust the effect of the user's particular mobility preferences and personal interests when selecting his/her trip mates (being even possible to consider only context-related features, such as origen, destination, routes, time and date).

4.3.2. Matching between Routes

Diverse metrics can be found in literature which are aimed at studying the similarity between moving object trajectories, which is a challenging task in many applications ranging from detection of moving patterns to inference of the future location of a moving object from its similar trajectories. In [21], Liu and Schneider adopt a distance function based on geographic features to detect similar trajectories. In [31], Vlachos *et al.* define a measure that is based on the Longest Common SubSequence (LCSS) model which gets an accurate and fast estimation of the distance between two trajectories even under noisy conditions. These metrics do not work well in our approach because our respective goals are totally different. We are not interested in detecting identical trajectories with similar moving patterns or estimating the future location of the vehicles. On the contrary, our focus is put on the origin and destination of the matched routes, or even on the existence of intermediate points within these routes that are very close to each other. To put it from another angle, two trajectories with identical origin and destination that include completely different intermediate points are not relevant for metrics like those proposed by Liu and Schneider and Vlachos *et al*. However, they are very significant in our approach, because they allow to identify a common itinerary for a ride-sharing opportunity. In order to fulfill our requirements, we have defined a metric based on Euclidean distances which is exploited by two different matching algorithms that deal with the routes traversed by the users.

1. Selection of a users' regular routes. The first algorithm is aimed at detecting regular routes of a particular user, by matching the information recorded in his/her profile about recent itineraries.

2. Selection of route for a shared ride. This algorithm decides whether two users can enjoy a shared ride by reckoning, among other features, the origins, destinations and intermediate points throughout their respective routes.

Both algorithms are certainly similar and combine GIS technologies to locate geographically the routes on a map with information about the users' mobility patterns, which have been modeled from the records of the GPS-style navigation equipment of their mobile terminals. Thanks to GIS technologies, the origins, destinations and intermediate points identified along each route[2] are geocoded and located on locally-recorded maps, prefetched from Google Maps whenever a WiFi connection is available, covering the users' routes.

In order to describe the algorithmic internals of both procedures, we model a user's route R_i as a sequence of consecutive locations in a two dimensional Euclidean space, that is:

$R_i =< (x_{1i}, y_{1i}), (x_{2i}, y_{2i}), \dots, (x_{ni}, y_{ni}) >$, being $O_{R_i} = (x_{1i}, y_{1i})$ and $D_{R_i} = (x_{ni}, y_{ni})$ its origin and destination, respectively.

Selection of a user's regular routes

When a user repeats several times an itinerary from an origin to a destination (or from a point close to this origin to a point close to this destination), this itinerary turns into a regular route. As shown in the pseudocode depicted in Alg. 1, the number of times considered is denoted by a configurable parameter α, whereas the origin and destination of the user's routes are compared to each other by adopting an Euclidean distance-driven metric (*dist*) and a threshold δ to detect if the matched points are close enough.

Selection of route for a shared ride

Our second matching algorithm decides whether the users U_1 and U_2 can travel together along a common itinerary, by considering their respective routes R_1 and R_2. If so, the algorithm returns the origin and destination of the shared ride.

As depicted in Alg. 2, our algorithm computes the Euclidean distance between the points included in the two routes in order to detect whether they are close enough. Our closeness criterion is driven by the aforementioned configurable tolerance threshold δ, which helps to decide whether the two users can travel together along the full route or just during part of the journey. The easiest case happens when the two users start from nearby origins and finish their itineraries in places that are also close to each other (scenario 1 in Figure 4). However, our algorithm also accommodates other scenarios aimed at analyzing the routes to detect overlapping thanks to the existence

[2] In our approach, the number and location of the intermediate points have been empirically adjusted via trial and error tests, by taking into account the length of each analyzed route and the results achieved with these settings. In particular, the approach that worked better adopts a higher frequency of points in the shortest routes (with average values of 17 intermediate points for itineraries with a length lower than 8 kms), and a proportionally lower number of intermediate locations in longer routes (e.g. 21 points for itineraries of about 28 km).

of a number of common points that are close enough (scenarios 2, 3 and 4). This way, it is possible that

- both users start the ride together and one of them drops off at the end of another's journey (scenario 2),
- one user is picked up in a meeting point and is dropped off en-route by the other one (scenario 3), and even (iii) both users arrive together to their respective destinations after sharing just a portion of the trip (scenario 4).

Really, the computational complexity of the matching process depends greatly on the amount of routes to be analyzed (and on the number of their intermediate points). The working of our algorithm helps to alleviate to some extent the negative effects of this concern. Specifically, when it comes to comparing two given routes, we start measuring the Euclidean distances between their respective origins and destinations. In case of these points are close enough (as per the threshold δ), the rest of intermediate locations can be discarded (since we have already detected that both itineraries are compatible for a shared ride, regardless of their respective intermediate points' locations). This saving in computations of Euclidean distances in some situations reinforces the computational feasibility of the matching process. Another view to look at has to do with the battery consumption, which typically becomes a big issue in execution environments involving mobile devices. In our domain, however, the matching algorithms run on the terminals of users who travel by car, where they can be connected to a power supply. Therefore, this is not the biggest issue.

Finally, having considered the spatial distribution of the points identified in each route, our algorithm incorporates extra matching criteria, such as the role offered by each user (driver/passenger), wait times, extra distances that users would be willing to tolerate in picking up and dropping off riders, and even walking distances to get meeting points. In case of detecting that the itineraries of two users are compatible for a shared ride, it is turn to reckon their interests and other mobility preferences towards a nice trip.

4.3.3. Matching between Users' Preferences

As introduced in Section 3, each user's device runs a matching algorithm to compare his/her personal preferences against those modeled in the profiles of other users, which are received over the vehicular ad-hoc network. Since we deal with a vehicular network without any infrastructure, we cannot rely on the computational resources available in a powerful centralized server when comparing the users' preferences. For that reason, we adopt a simple context-aware algorithm to compute the matching level of Eq. (1), where we measure the resemblance between two users U_1 and U_2 (denoted by $matching(U_1, U_2)$) by combining two components that are weighted by a configurable factor γ (with γ ∈ [0, 1]). The first component compares the *personal preferences* of both users, while the second one considers only their respective *mobility needs* (denoted by $matching_{PP}(U_1, U_2)$ and $matching_{MN}(U_1, U_2)$, respectively).

$$matching(U_1, U_2) = \gamma \cdot matching_{PP}(U_1, U_2) + (1 - \gamma) \cdot matching_{MN}(U_1, U_2) \qquad (1)$$

The two components reckon features referred both to particular interests (denoted by f_{Pk}) and to mobility-related requirements (f_{Ml}), so that the greater the number of common features, the higher the resulting matching level. This idea is behind Eq. (2) where $sim(U_1, U_2, f_{Pk})$ will be 1 when the

Algorithm 1. Our algorithm for identification the user *U*'s regular routes

```
Input: Routes traversed by the user $U$ (denoted by $R_U = \{R_1, R_2, ..., R_N\}$, $O_{R_i}$ and $D_{R_i}$ being the origin and destination of route $R_i$, respectively)
Output: Regular routes identified for the user $U$ (denoted by $RR_U$)
procedure REGULAR_ROUTES_IDENTIFICATION($R_U$)
    for $i$ do ← $1, N$
        $counter_{R_i} = 1$
    end for
    for $i$ do ← $1, N$
        for $j$ do ← $i + 1, N$
            if (($dist(O_{R_i}, O_{R_j}) <= \delta$) AND ($dist(D_{R_i}, D_{R_j}) <= \delta$)) then
                $counter_{R_i} ++$
            end if
        end for
        if ($counter_{R_i} >= alpha$) then
            $add(RR_U, R_i)$
        end if
    end for
    return $RR_U$
```

Algorithm 2. Algorithm for selecting a route for a shared ride between U_1 and U_2

```
Input: Routes traversed by the users $U_1$ and $U_2$ ($R_{U_1}$ and $R_{U_2}$, respectively):
        $R_{U_1} =< (x_{11}, y_{11}), (x_{21}, y_{21}), \ldots, (x_{n1}, y_{n1}) >$
        $R_{U_2} =< (x_{12}, y_{12}), (x_{22}, y_{22}), \ldots, (x_{m2}, y_{m2}) >$
Output: Origin and destination of the shared ride.
procedure ROUTE_SELECTION_FOR_SHARED_RIDE($R_{U_1}$, $R_{U_2}$)
    for $j$ do $\leftarrow 1, m$
        for $i$ do $\leftarrow 1, n$
            if $(dist((x_{j2}, y_{j2}), (x_{i1}, y_{i1})) <= \delta)$ then
                $add(Aux, (x_{i1}, y_{i1}))$
            end if
            if $(Aux.size() > 1)$ then
                $(x_{ST}, y_{ST}) = Select\_Closest\_Point(Aux)$
                $add(ST, (x_{ST}, y_{ST}))$        ◃ ST comprises $R_1$'s points closest to $R_2$.
            else
                $ST = Aux$
            end if
        end for
    end for
    return $(ST[0], ST[ST.size() - 1])$
                                        ◃ 1st and last elements in ST are origin and destination of the shared route.
```

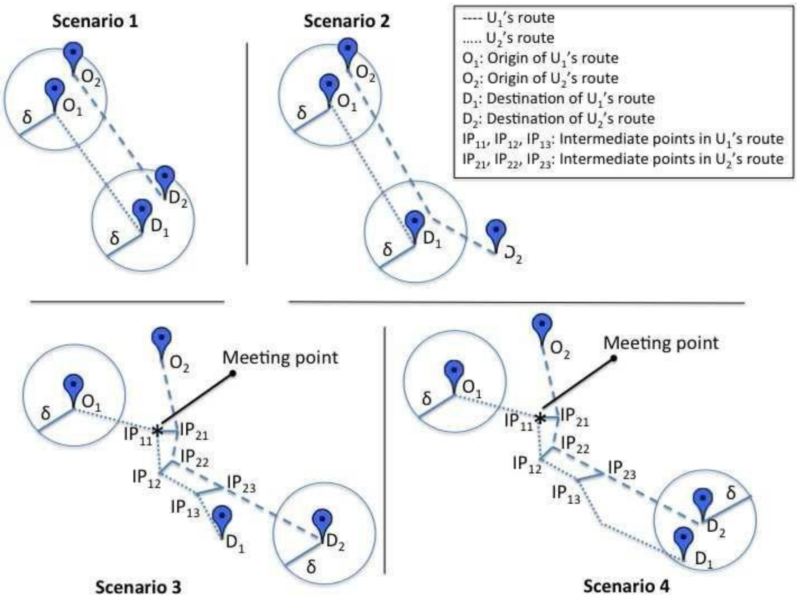


Figure 4. Some scenarios supported by our matching algorithm between users' routes

k-th feature f_{P_K} takes the same value in the profiles of both users (e.g. if U_1 and U_2 like sports), and 0 otherwise. The same is valid for the mobility requirements in Eq. (3), so that $sim(U_1, U_2, f_{Ml})$ will be 1 when the two users prefer to avoid tolls in their shared trips (assuming that f_{Ml} is referred to "toll avoidance"). In both equations, N_1 and N_2 refer to the number of features of each type (interests and mobility preferences) that are considered when matching the potential riders' preferences.

$$matching_{PP}(U_1, U_2) = \frac{1}{N_1}\sum_{K=1}^{N1} Sim(U_1, U_2, fP_K) \qquad (2)$$

$$Sim(U_1, U_2, fP_K) = \begin{cases} 1 \; if \; fP_K \; takes \; the \; same \; value \; for \; U_1 \; and \; U_2 \\ 0 \; otherwise \end{cases}$$

$$matching_{MN}(U_1, U_2) = \frac{1}{N_2}\sum_{K=1}^{N_2} Sim(U_1, U_2, fM_l) \qquad (3)$$

$$Sim(U_1, U_2, fM_l) = \begin{cases} 1 \; if \; fM_l \; takes \; the \; same \; value \; for \; U_1 \; and \; U_2 \\ 0 \; otherwise \end{cases}$$

By combining the above equations, our algorithm considers that two users are like-minded riders if their matching level is greater than a configurable threshold. The value of this threshold allows to tune up the matching process, ranging from a soft comparison (where the user and his/her trip mates would share just some preferences) to a stricter process (where most of the user-defined features must be necessarily satisfied by his/her trip mates).

4.4. Application layer

The multi-layer architecture of our approach allows the smart VANET to be easily reused in the development of other context-aware functionalities, beyond ride-sharing schemes. For that purpose, most of the capabilities accommodated in distribution and ad-hoc communications layers would not need to be changed at all, whereas the users profiles, matching algorithms and interfaces provided at application layer would require some modifications. Specifically, the application level includes interfaces that to enable the user to enjoy the functionalities described so far in our ride-sharing platform, along with security mechanisms aimed at the proper identification of trip mates and the provision of guarantees and compensations in case of agreement violations from any party. In particular, the secure identification of the riders relies on mechanisms based on asymmetric crytography, whereas the provision of guarantees and compensations resorts to private communication channels to allow the users to get in contact to each other and agree different solutions in case of no-shows. The technical details of the algorithms related to asymmetric cryptography and the infrastructure of the call center are out of the scope of this paper.

5. An Example Scenario

Peter is 25 years old and he has started to work in a textile factory two weeks ago, which is 15 kilometers away from the coast village where he lives. Peter is aware of the large expenses derived from his daily round-trip, leaving home at 8:00 a.m. and coming back at about 6:30 p.m. Peter does not feel still at ease with his new job mates and does not dare to propose them a shared ride to save expenses.

While having a beer with friends, Bob talks to Peter about a new application that automatically tracks your itineraries, discovers frequent mobility patterns and proactively organizes shared trips with like-minded users along your routes of interest. After downloading and installing the application, Peter logs into and defines some personal preferences about his possible trip mates for a shared ride: he does not care about gender, age and smoking habits, but he prefers taking the role of driver and travelling with people who like talking about sports and animals.

On Saturday in the morning, Peter receives a call from his friends to go playing a basketball match. Peter drives his car along a lonely secondary road and finally enters the city throughout a commercial area with heavy traffic. Then, his smartphone joins a sporadic ad-hoc network established among the handheld devices owned by the occupants of close cars. Having previously detected the round trip that Peter makes daily to go working and come back home, the application triggers then the search for potential riders.

In that moment, Anne is entering in a garage (for a routing inspection of her car) close to the traffic-lights where Peter is stopped. The matching algorithms confirm that Peter's routes could be interesting for Anne because they include places where she was recently. Specifically, the last week Anne went several days to a University very close to the industrial area where Peter works, where she is enrolled in a Computer Science course. Her preferences seem to match perfectly with Peter's ones, because her profile defines non-smoking young men as favourite trip mates. After being notified by the application, Anne accepts the possibility of sharing expenses with Peter every morning to attend her classes at the University. Her confirmation is sent to Peter's device. Then, he checks that Anne's avatar is a picture of a tender litter of cats, from which he infers that she also loves animals. Consequently, Peter opts for her as mate in the mornings to share 5 daily rides per week in a long term. Upon the decision, Anne and Peter receive their corresponding confirmations over the ad-hoc network.

6. Experimental Evaluation

In order to validate our proposal, we have organized our tests as follows:

- Our first goal was to evaluate the opinion of a set of users about the concept of proactively-organized shared rides, and about the working of the algorithms adopted for matching preferences and itineraries of the prototype developed for testing purposes, as we will describe in Section 6.1.
- The second part of our tests was aimed to assess the technical refinements that have been specifically designed in the distribution layer and in the ad-hoc communications level, which are crucial in the successful deployment of our ride-sharing platform in a vehicular environment. For that purpose, we have developed a VANET simulator (see details in [4]) to assess the performance of our VaNetLayer. Details will be given in Section 6.2.

Before describing in detail each phase of our experiments, it is worthy to note that we have not carried out a formal evaluation of our matching algorithms (including, for instance, a comparison with other known approaches) because these components are not the real cornerstone of our research work. To put it from another angle, this paper presents a possible application of a smart VANET that is joined to the proactive organization of ride-sharing opportunities. Therefore, we require a "light"[3] matching algorithm

[3] Note the computational restrictions of handheld devices.

for comparing the routes of the users. However, our research goal is to validate the working of the VANET as a reliable communication environment (even beyond the application domain explored in this paper). For that reason, we have carried out an informal evaluation of the matching algorithms driven by questionnaries, along with a more technical experimentation aimed at comparing the performance of the refinements we have envisaged in the virtualization layer against traditional approaches.

6.1. Evaluation driven by interaction between users and a prototype

We have made field experiments to evaluate the appraisal of the concept by potential end users. To this aim, we developed a prototype of the system that was used by 46 volunteers from the community of students, academic staff and administrative staff of the University of Vigo, which has premises in the cities of Vigo, Pontevedra and Ourense. The prototype is based on a mobile app that is executed on the devices that we lent to the participants.

The volunteers used the system during 3 weeks. The first week was devoted to building up each one's profile locally, and the system provided recommendations during the subsequent fortnight. Finally, we requested them to fill out a questionnaire to give their opinion about diverse aspects of the experience. The users were asked about (i) whether they knew their trip mates before our experiments, (ii) the precision of our algorithms when detecting commonly traversed itineraries and matching mobility needs of different users, (iii) the relevance of the personal preferences when identifying trip mates, and (iv) their overall impression about our ride-sharing prototype.

A total of 98 carpooling opportunities were identified, out of which 72 were finally accepted by the participants. Out of them, most of the rides involved 3 or 4 trip mates who had never met previously. The fact that our prototype was able to gather together a group of strangers for a shared trip confirms the potential of a smart VANET towards the proactive organization of ride-sharing opportunities. Regarding the influence of personal preferences, the participants revealed that the particular interests of other users are mainly relevant when selecting trip mates for a long itinerary (e.g. as from Pontevedra to Vigo or the other way round, where both cities are 28 km away); on the contrary, for short trips (e.g. as from different parts of Vigo to the campus, which is located in the outskirts), the participants put the focus on the convenience of the routes instead of prioritizing the identification of like-minded riders.

Even though we deal with a very reduced sample to draw statistically significant results, 85% of the participants highlighted the precision of the matching algorithms adopted to identify both own regular routes and common itineraries to be shared with other riders. In this regard, most of the finally accepted carpooling opportunities involved users who were picked up in an agreed meeting point and dropped off en-route by the driver (just like in the scenario 3 depicted in Figure 4), whereas just in 10% of cases the users started the trip from close origins and arrived their respective destinations together (scenario 1). Regarding the overall impression of the users about our ride-sharing application, 65% of the participants rated the experience "positively" or "very positively"; 26% had a "neutral" opinion and only the remaining 9% said that they had found the concept "uninteresting". Besides, we could confirm our expectations regarding where and when the communications would take place, inasmuch as 85% of the opportunities were identified in the vicinity of the three campuses, predominantly during the first hour of the working day (when there are more cars arriving) and during lunch time. Out of the remaining 15%, more than half of the recommendations were triggered opportunistically during working days in the city centres, whereas the rest would occur during the weekend in leisure places (malls, discos, beaches or thermal areas). The most significat results from our experiments are summarized in Table 1.

6.2. Evaluation driven by a VANET simulator

In order to assess the technical viability of the proposal on a greater scale, we have made simulation experiments with the smart VANET to characterise relevant application-level metrics against greater numbers of users. In this section, we present the results of the simulation experiments we have carried out to assess the benefits of the mechanisms we have put into the VaNetLayer and the refinements envisaged in the routing protocol VNAODV+. To this aim, we focus on a comparison with the VNLayer and with the protocols AODV and VNAODV, looking at their performance in terms of network-layer and virtualization-layer metrics with simple constant bitrate (CBR) flows of communication between random pairs of nodes.

Our simulation environment combines the SUMO[4] simulator of urban mobility and the ns-3 network simulator. On the one hand, SUMO provided realistic mobility traces for a varying number of vehicles on the streets of an urban area of 476 x 476 meters from downtown Cuenca (Ecuador), previously captured in OpenStreetMap. On the other hand, ns-3 was used to simulate communications with IEEE 802.11p PHY/MAC, assuming wireless signals propagating according to the shadowing radio model and a maximum transmission range of 250 meters.

The number of communication sessions was initially set to 10, although this parameter was modified with the goal of analyzing its influence on some of the metrics adopted in our experiments. These 10 CBR flows were set to transmit 500KB per second and to last throughout the simulation time. Each simulation lasted 450 seconds, but the traces for the first

[4] http://sumo.sourceforge.net/

Table 1. An overview of the results extracted from the users' responses to our questionnaires

Number of ride-sharing opportunities triggered opportunistically in the VANET	98
Accepted shared rides ratio	73.5%
Opinion about matching algorithms:	
Positive ratings ratio	85%
Neutral ratings ratio	13%
Negative ratings ratio	2%
Opinion about our praoctively-organized carpooling opportunities:	
Very positive & positive ratings ratio	65%
Neutral ratings ratio	26%
Negative ratings ratio	9%

50 seconds in each simulation were skipped to ensure stable measurements. We repeated each simulation 10 times for each data point collected. Besides, VaNet-Layer as cornerstones of the virtualization layer, and (ii) AODV, VNAODV and VNAODV+ as routing protocols that work about it. Within these settings, we looked at the following metrics:

- Average duration of VN downtimes, which is related to periods during which there are vehicles (PNs) in a region but no node is acting as leader (recall Section 4.1).
- Virtualization overhead, related to the number of messages that VNLayer/VaNetLayer and the routing protocols exchange among the vehicles.
- Packet delivery ratio, that is, the ratio between the number of packets delivered to the destinations and the number of packets sent by the sources.
- End-to-end delay, which identifies the average time it takes for packets to reach their destinations after they are sent by the sources.

The average duration of VN downtimes with the VNLayer and the VaNetLayer is shown in Figure 5 against varying numbers of physical nodes (60 to 160 vehicles). The number of communication sessions was initially set to 10. It can be seen that the VaNet-Layer greatly reduced the duration of the downtimes by around 10% in comparison with the VNLayer, thus favouring greater availability of the virtual nodes.

Figure 6 represents the variation of the virtualization overhead caused by the VNLayer and the VaNet-Layer, again with 10 communication sessions. It can be seen that the VaNetLayer achieves substantial savings and accommodates greater numbers of PNs in a more scalable manner.

Regarding the comparison of the routing protocols, Figures 7 and 8 represent the packet delivery fractions achieved by AODV, VNAODV and VNAODV+, against a varying number of PNs (with 10 communication sessions going on) and a varying number of communication sessions (with 120 PNs). It is clear from both graphs that VNAODV+ outperforms the other protocols, since it achieved 20% to 35% better delivery ratios. The fact that VNAODV performs significantly better than AODV reinforces the idea that it is beneficial to lean on a we worked with TCP as transport protocol and compared the workings of (i) the VNLayer and our

virtualization layer, while the differences between VNAODV+ and VNAODV prove the advantages of our refinements.

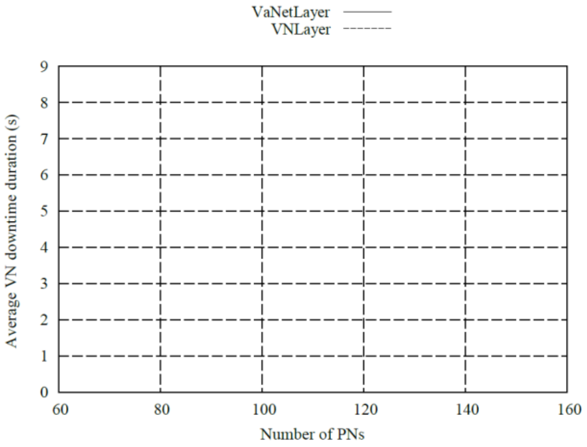


Figure 5. Average duration of VN downtimes against number of physical nodes

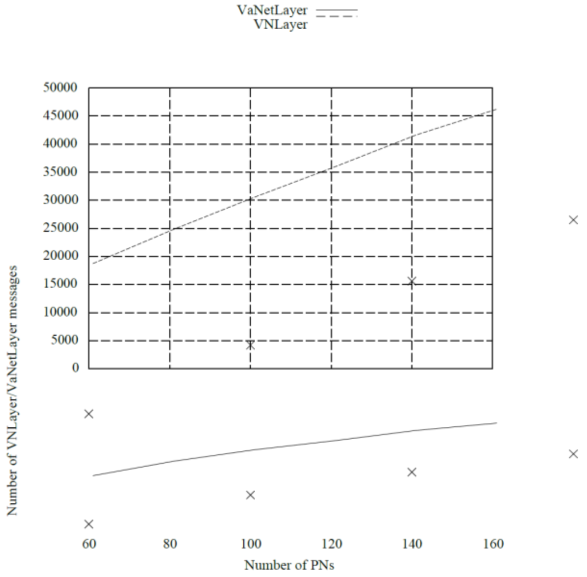
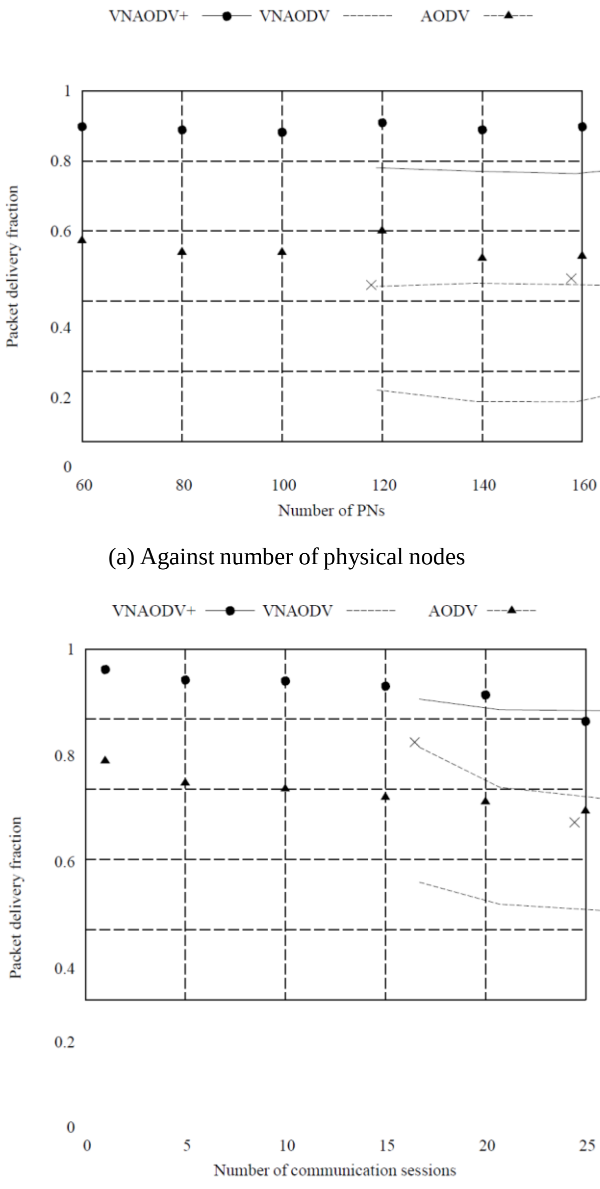


Figure 6. Virtualization overhead against number of physical nodes

Figure 8(a) shows that, with 10 communication sessions going on, the delivery latencies with VNAODV+ were significantly lower than those provided by VNAODV and AODV, entailing that our proposal does not only deliver more packets, but also delivers them faster. Looking at scenarios with more communication sessions in place, we notice in Figure 8(b) that VNAODV+ yields greater latencies when faced with greater traffic loads, even though it remains consistently better than VNAODV. The line corresponding to AODV remains fairly constant and below the one corresponding to with 20 and 25 sessions. However, it is important to remember that AODV delivers less than 60% of the packets, whereas VNAODV+ delivers at least 20% more of them. Many of those additional packets reach their destinations after surviving mishaps (e.g. collisions, route breakages, leader withdrawals, ...) but the excess delay does not yet surpass a few round-trip times, so those packets may still be useful for most applications.

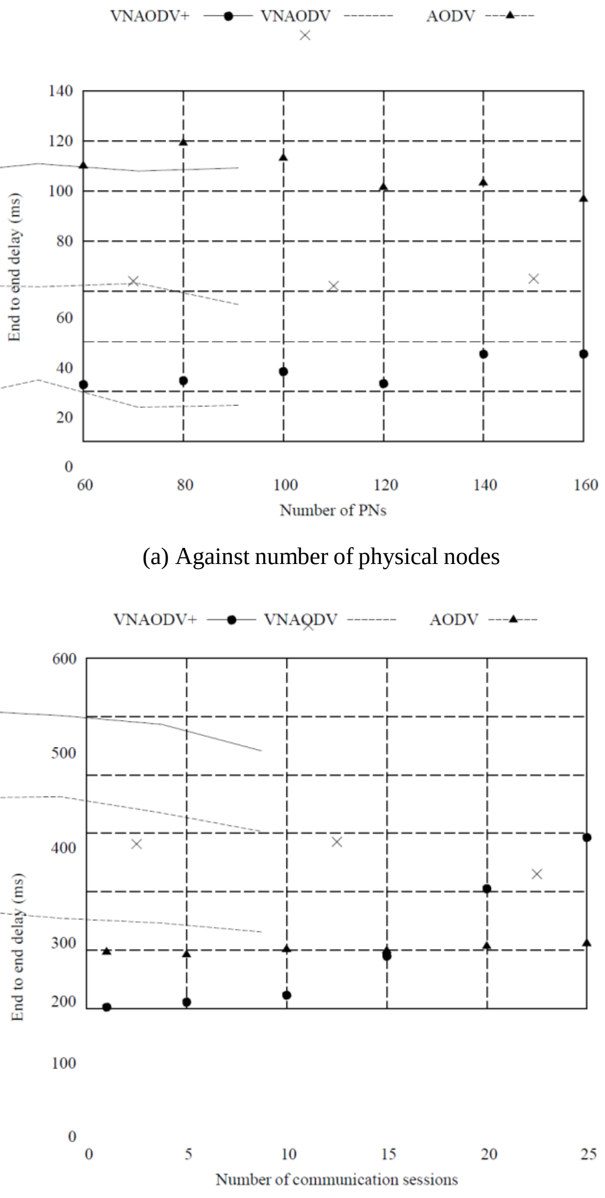
Our experiments also enabled to characterise the cross-relationships between a number of parameters, including (i) the number of connected vehicles, (ii) the areas covered by the smart VANETs, (iii) the number of transactions completed by the distribution agents and the number of uncompleted ones, and (iv) the devices' battery consumption. In this regard, probably the most important observation for this paper is that the load imposed by our application does not grow exponentially with vehicle density and with the number of connected devices, inasmuch as the search for potential trip mates is sequenced by priorities and terminates once a sufficient number of candidates has been found. Therefore, we do not cause



(a) Against number of physical nodes

(b) Against number of communication sessions

Figure 7. Variation of packet delivery ratio



(a) Against number of physical nodes

(b) Against number of communication sessions

Figure 8. Variation of end-to-end delay

either excessive burden on the communications layer or strain on the batteries (which take also advantage of the fact that mobile devices can be plugged in while the users are traveling by car).

Additionally, it is important to note that the vehicles typically remained connected to the smart VANETs for much longer than the time needed to receive the results of the group matchings after a distribution agent initiates one transaction (seconds vs tens of milliseconds). Thus, in our experiments, at least 90% of the transactions initiated by the distribution agents could always be completed, meaning that the results of the group matchings were received by the corresponding agents in time.

7. Conclusions and Further Work

We have described a platform that finds solutions to proactively serve the users' mobility needs without their active involvement, which is a novel feature in existing ride-sharing approaches that contributes to promote this kind of initiatives. Our approach expands the possibilities of organizing appealing shared rides beyond the specific settings where the idea has traditionally worked (e.g. among people who happen to work in a same industrial area). To this aim, we take advantage of (i) the existence of heavily trafficked zones –where people with compatible mobility needs could happen to be– and (i) the fact that most of the occupants of these vehicles typically take with them a mobile device with capabilities for automatically keeping track their oftenly-traversed routes. Besides the technological capabilities of these terminals, our approach exploits the users' context (related to their mobility patterns and even personal preferences) and a set of refinements that we have envisaged to support reliable ad-hoc communications over the vehicles of the VANET (by emulating static virtual nodes by mobile physical nodes). Likewise, considering the computational restrictions of mobile devices, our approach also contributes with mechanisms to allocate over the terminals connected to the VANET the communications and computations required in the organization of ride-sharing opportunities.

The proper operating of our approach has been experimentally validated by combining a VANET simulator with a prototype which 46 participants were interacting with during 3 weeks. An informal evaluation driven by questionnaires confirmed that most of the users enjoyed our concept of ridesharing opportunities triggered opportunistically in a VANET, revealing positive opinions about the algorithms adopted in the proactive identification of potential trip mates. Regarding the simulations, the measures obtained served to check that our refinements in the ad-hoc communication layer of the VANET outperform previous approaches, which have been unaware of the pecularities of the vehicular communication environment explored in our research. These improvements have been quantified in terms of packet delivery fraction and average end-to-end delay, duration of downtimes of the virtual nodes supported by the cars on move, and virtualization overhead related to routing of data over the VANET.

Regarding the future work, we plan to support scenarios in which exact time and/or precise destination are actually less important than the activities to do at the destination, or the likings, interests and needs of the people who would travel together. This approach would be useful, for example, for people who want to socialise in their spare time (e.g. to meet people who would like to go to latin dance clubs on a Friday night, or shopping on a Saturday afternoon), or for people who want to attend mass events (e.g. a concert or a football match). The goal is to make it possible to establish group agreements, for example, to organize a ride conditional on gathering a sufficient number of passengers before a given deadline. Providing these functionalities requires contextaware group recommendation strategies, beyond the locally-executed matching algorithms that we have been developed for an ad-hoc network without any pre-established infrastructure.

Acknowledgment

This work has been funded by the Ministerio de Educación y Ciencia (Gobierno de España) research project TIN2013-42774-R.

References

[1] **N. Agrawal, S. S. Verma.** Multilevel security model using distributed keys in MANET. *International Journal of Computer Applications*, 2013, Vol. 70, No. 2, 10– 18.

[2] **S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, H. Zedan.** A comprehensive survey on vehicular Ad-Hoc network. *Journal of Network and Computer Applications*, 2014, Vol. 37, Issue 1, 380-392.

[3] **J. F. Bravo-Torres, M. López-Nores, Y. Blanco-Fernández**. Virtualization support for complex communication in Vehicular Ad-Hoc Networks. *MTA Review*, 2013, Vol. 23, No. 2,121–140.

[4] **J. F. Bravo-Torres, M. López-Nores, Y. Blanco-Fernández, J. J. Pazos-Arias**. On the use of virtual mobile nodes with real-world considerations in vehicular ad-hoc networks. In: *9th International Conference on Communications (COMM)*, 2012, pp. 193–196.

[5] **F. Borrego-Jaraba, I. Luque-Ruiz, M. Gomez-Nieto.** A NFC-based pervasive solution for city touristic surfing. *Personal and Ubiquitous Computing*, 2011, Vol. 15, No. 7, 731–742.

[6] **S. Choudhury, K. Salomaa, S. G. Akl.** Cellular automaton-based algorithms for the dispersion of mobile wireless sensor networks. *International Journal of Parallel, Emergent and Distributed Systems*, 2014, Vol. 29, No. 2, 147–177.

[7] **B. Chowdhury, M. U. Chowdhury.** RFID-based real time smart waste management system. In: *Australasian Telecommunication Networks and Applications Conference (ATNAC)*, 2007, pp. 175–180.

[8] **D. Chowdhury, N. Nedic, V. Dimitrieski.** Realtime carpooling and ride-sharing: Position paper on design concepts, distribution and Cloud Computing strategies. In: *Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 781–786, 2013.

[9] **S. Dolev, S. Gilbert, N. A. Lynch, E. Schiller, A. A. Shvartsman, J. L. Welch.** Virtual mobile nodes for mobile ad-hoc networks. In: *18th Annual Conference on Distributed Computing (DISC)*, 2004, pp. 230–244.

[10] **A. Domingo, B. Bellalta, M. Palacin, M. Oliver, E. Almirall.** Public open sensor data: revolutionizing Smart Cities. *IEEE Technology and Society Magazine*, 2013, Vol. 32, No. 4, 50–56.

[11] **T. Frantti, M. Majanen**. An expert system for real-time traffic management in wireless local area networks. *Expert Systems with Applications*, 2014, Vol. 41, No. 10, 4996–5008.

[12] **M. Furuhata, M. Dessouky, F. Ordonez, M. Brunet, X. Wang, S. Koenig**. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 2013, Vol. 57, 28–46.

[13] **M. Gerla, L. Kleinrock**. Vehicular networks and the future of the mobile Internet. *Computer Networks*, 2011, Vol. 55, No. 2, 457–469.

[14] **H. Hartenstein, K. P. Laberteaux.** A tutorial survey on Vehicular Ad-Hoc Networks. *IEEE Communications Magazine*, 2008, Vol. 46, No. 6, 164–171.

[15] **J. Jakubiak, Y. Koucheryavy.** State of the art and research challenges for VANETs. In: *5th IEEE Consumer Communication and Networking Conference (CCNC)*, 2008, pp. 912–916.

[16] **A. P. Janani, M. Sakthivel, M. Saravanan.** AODV_HPR: A high power Ad-Hoc routing for highly mobile short time military communication applications. *International Journal of Engineering and Technology*, 2013, Vol. 5, No. 5, 3781–3793.

[17] **J. M. Su, C. F. Huang.** An easy-to-use 3D visualization system for planning context-aware applications in smart buildings. *Computer Standards & Interfaces*, 2014, Vol. 36, No. 2, 312–326.

[18] **V. Kostakos, T. Ojala, T. Juntunen.** Traffic in the Smart City: exploring city-wide sensing for traffic control center augmentation. *IEEE Internet Computing*, 2013, Vol. 17, No. 6, 22–29.

[19] **F. Li, Y. Wan.** Routing in Vehicular Ad-Hoc Networks: a survey. *IEEE Vehicular Technology Magazine*, 2007, Vol. 2, No. 2, 12–22.

[20] **Y. Lin, Y. Chen, S. Lee.** Routing protocols in Vehicular Ad-Hoc Networks: a survey and future perspectives. *Journal of Information Science and Engineering*, 2010, Vol. 26, No. 3, 913–932.

[21] **H. Liu, M. Schneider.** Similarity measurement of moving object trajectories. In: *3rd ACM SIGSPATIAL International Workshop on GeoStreaming (IWGS)*, 2012, pp. 19–22.

[22] **M. López-Nores, Y. Blanco-Fernández, J. J. Pazos-Arias, J. F. Bravo-Torres.** Leveraging ad-hoc networking and mobile cloud computing to exploit short-lived relationships among users on the move. In: *International Conference on Intelligent Cloud Computing (ICC)*. To be published, 2014.

[23] **H. Moustafa, Y. Zhang.** Vehicular networks: techniques, standards and applications. *CRC Press*, 2009.

[24] **T. Nadeem, S. Dashtinezhad, C. Liao, L. Iftode.** TrafficView: Traffic data dissemination using car-to-car communication. *ACM Mobile Computing and Communications Review*, 2004, Vol. 8, No. 3, 6–19.

[25] **A. Nandan, S. Das, S. Tewari, M. Gerla, L. Klienrock**. AdTorrent: Delivering location cognizant advertisements to car networks. In: *Proceedings of International Conference on Wireless On Demand Network Systems and Services*, 2006, pp. 203–212.

[26] **A. Nandan, S. Dan, B. Zhou, G. Pau, M. Gerla**. AdTorrent: Digital billboards for vehicular networks. In: *Proceedings of Workshop on Vehicle-to-Vehicle Communication*, 2005, pp. 7–18.

[27] **S. Shah, B. Beferull-Lozano**. Joint sensor selection and multihop routing for distributed estimation in Ad-Hoc Wireless Sensor Networks. *IEEE Transactions on Signal Processing*, Vol. 61, No. 24, 6355-6370, 2013.

[28] **Z. Siddiqi, R. Buliung.** Dynamic ridesharing and information and communications technology: Past, present and future prospects. *Transportation Planning and Technology*, 2013, Vol. 36, No. 6, 479–498.

[29] **Y. Toor, P. Muhlethaler, A. Laouiti.** Vehicle Ad-Hoc networks: applications and related technical issues. *IEEE Communications Surveys & Tutorials*, 2008, Vol.10, No. 3, 74–88.

[30] **S. Trifunovic, B. Distl, D. Schatzmann, F. Legendre.** WiFi-opp: ad-hoc-less opportunistic networking. In: *ACM workshop on Challenged networks (CHANTS)*, 2011, pp. 37–42.

[31] **M. Vlachos, G. Kollios, D. Gunopulos**. Discovering similar multidimensional trajectories. In: *18th International Conference on Data Engineering (ICDE)*, 2012, pp. 673–684.

[32] **H. Wirtz, T. Heer, R. Backhaus, K. Wehrle.** Establishing mobile ad-hoc networks in 802.11 infrastructure mode. In: *ACM workshop on Challenged networks (CHANTS)*, 2011, pp. 49–52.

[33] **L. Wischhof, A. Ebner, H. Rohling.** Information dissemination in self-organizing intervehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, 2005, Vol. 6, No. 1, 90–101.

[34] **J. Wu.** A simulation study on using the virtual node layer to implement efficient and reliable MANET protocols. *Ph.D. Thesis*, http://groups.csail.mit.edu/tds/papers/Wu/JiangWu-ThesisFinal.pdf, 2011.

[35] **P. Zhou, T. Nadeem, P. Kang, C. Borcea, L. Iftode**. EZCab: a cab booking application using short-range wireless communication. In: *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, 2005, pp. 35–46.

Received October 2014.