Dear Mobile Agent, Could You Please Find Me a Parking Space?

Oscar Urra(✉) and Sergio Ilarri

Department of Computer Science and Systems Engineering, I3A,
University of Zaragoza, Zaragoza, Spain
`ourra@itainnova.es`, `silarri@unizar.es`

Abstract. Vehicular ad hoc networks (VANETs) have attracted a great interest in the last years due to their potential utility for drivers in applications that provide information about relevant events (accidents, emergency brakings, etc.), traffic conditions or even available parking spaces. To accomplish this, the vehicles exchange data among them using wireless communications that can be obtained from different sources, such as sensors or alerts sent by other drivers. In this paper, we propose searching of parking spaces by using a mobile agent that jumps from one vehicle to another to reach the parking area and obtain the required data directly. We perform an experimental evaluation with promising results that show the feasibility of our proposal.

Keywords: Vehicular networks · Mobile agents · Distributed query processing · Data management · Parking spaces

1 Introduction

Vehicular networks (VANETs) [7] are mobile ad hoc networks where the vehicles can establish connections among them by using short-range wireless communications (such as IEEE 802.11p [11]) and exchange data that can be interesting for the drivers, such as information about traffic jams, accidents, or scarce resources such as available parking spaces. However, deploying applications that retrieve and exploit those data is not an easy task [4], since the vehicles are continuously moving and the interval of time for exchanging data is very short.

The data exchanged by the vehicles can be stored locally on them and a driver that wants to retrieve information can submit a query over those data by following different approaches: (1) push-based query processing assumes that the relevant data are proactively pushed into the vehicles [1], so the query can be processed locally by exploiting the previously received data; (2) pull-based query processing implies disseminating the query in the network in such a way that a number of vehicles are explicitly asked about data that could be relevant for the query. Push-based approaches are easier to deploy and simpler, but the potential queries are constrained by the data that are actively exchanged.

On the other hand, we believe that mobile agents could be suitable for distributed query processing in VANETs. Mobile agents [8] are programs that have

M. Kirikova et al. (Eds.): ADBIS 2017, CCIS 767, pp. 82–90, 2017.
DOI: 10.1007/978-3-319-67162-8_10

the capability to pause their execution, move to another computer, and resume their execution. In this way, they can locate the relevant data sources and move there to process the data locally and filter out the irrelevant information, instead of sending the collected data to a central location, which may not be possible in a VANET due to the limitations and high costs of mobile networks.

In this paper, we study the potential use of mobile agents to retrieve data about available parking spaces in a city. The structure of the paper is as follows. In Sect. 2, we describe an approach to solve the problem by using mobile agents. In Sect. 3, we analyze a use case scenario to retrieve information about available parking spaces, and present an experimental evaluation. In Sect. 4, we present some related work and, finally, in Sect. 5 we show our conclusions and outline some prospective lines of future work.

2 Retrieving Parking Spaces Using Mobile Agents

We consider a scenario where vehicles exchange data about nearby available parking spaces by using a push-based data sharing approach. However, push-based approaches only disseminate popular data that are expected to be relevant in a nearby area, based on the evaluation of spatio-temporal criteria. Therefore, a pure push-based approach cannot handle situations where a driver is interested in retrieving information about available parking spaces located in further areas. Instead, we need to explicitly disseminate a query (pull-based approach) to retrieve data stored by vehicles located near the destination area. For that purpose, a mobile agent can autonomously jump from car to car to reach the relevant data sources that store data about those parking spaces (i.e., vehicles near the destination area) and query them, following the steps described in [10]:

1. A mobile agent is created that will reach the destination area by jumping from car to car, by using only ad hoc short-range wireless communications.
2. The agent retrieves data about available parking spaces by querying the local databases available in vehicles inside the destination area.
3. Once the agent has retrieved enough data about available parking spaces, it returns to the vehicle of the driver interested in those parking spaces (again, by hopping from vehicle to vehicle) and provides the results collected.

3 Experimental Evaluation

In this section, we first describe the use case scenario that we propose and then we perform a number of experiments to evaluate the performance of our proposal. We repeated each experiment 50 times (with different random starting positions for the vehicles and different trajectories) and we report the average results.

3.1 Experimental Setup

In the use case considered, a person traveling along a highway approaches a city, and when he/she is 15 km from the city he/she wants to obtain information about available parking spaces near the city center. Then, a mobile agent is launched, that travels to that area to collect information about the parking spaces by querying the vehicles present there, and will return to the driver information about a specific number of parking spaces. For the evaluation, we have used the MAVSIM simulator [9], that allows the simulation of both mobile agents and traffic in a realistic way (it uses road maps extracted from OpenStreetMaps and simulates the blocking of signals by buildings using the method described in [6]).

The map scenario chosen (Fig. 1) is a portion of the city of Zaragoza (Spain) and some fragments of interurban roads where the driver asking for parking spaces is located when the experiment starts. We consider an *urban area*, where the speed of vehicles is 50 km/h and buildings block the wireless signals, and a *interurban area*, where vehicles can travel up to 120 km/h, and there are no buildings blocking radio signals. The rectangle shown in Fig. 1 shows the destination area (the city center), whose size is 0.25 km^2. Some parameters of the simulations are shown in Table 1, and a few of them deserve further explanations:

(a) Selected roads (b) Whole area in OpenStreetMaps

Fig. 1. Scenario map used in the simulation

- The *hop strategy* refers to the algorithm used by the mobile agent when it travels by hopping among vehicles and it must choose, among several candidates, the most promising to reach the agent's final destination. We have chosen a greedy heuristic (called *map distance*) that uses the remaining distance (the shortest path) to the destination and has a good performance [10].
- The *mobile agent's hop delay* is the time needed by the mobile agent to hop from a vehicle to another one within the communication range. After performing experiments with real mobile devices [10], we decided not to assume a best-case scenario and establish the travel time, pessimistically, as one second.
- The default *percentage of vehicles with relevant data* is set to 50%, which means that only half of the vehicles have useful data for the query processing. This percentage allows to simulate cases where some vehicles do not participate in the data sharing of information about parking spaces.

Table 1. Simulation parameters

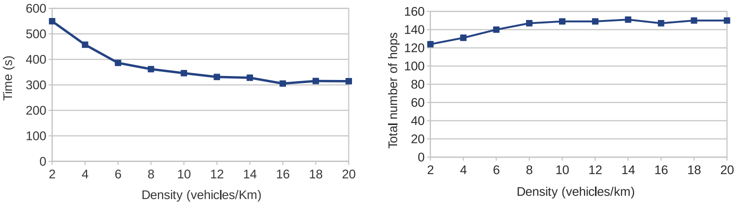
Parameter	Default value
Map dimensions	4 × 4 Km (urban area)
	20 Km (interurban area)
Map scenario	Zaragoza (Spain) and its surroundings
Size of the destination area	$0.25\,\mathrm{Km}^2$
Distance to the destination area	15000 m
Density of vehicles	50 vehicles/Km^2 (urban area)
	10 vehicles/Km (interurban area)
Speed of the vehicles	50 Km/h ± 10% (urban area)
	120 Km/h ± 15% (interurban area)
Hop strategy	MAP (map distance)
Percentage of vehicles with relevant data	50%
Total number of parking spaces in the destination area	220 *parking spaces*
Percentage of available parking spaces per vehicle	10%
Data processing delay	5 s
Number of available parking spaces to retrieve	10 *available parking spaces*
Parking occupancy	90%
Data collection and warning timeouts	4 and 3 min respectively
Communication range and mobile agents' hop delay	250 m and 1 s respectively

- The *percentage of available parking spaces per vehicle* is the percentage of available parking places within the area that are stored in a vehicle. We have chosen a default valor of 10% for this parameter, which is quite pessimistic. This means that, if there are 50 available parking spaces, each vehicle with data would have information about 5 of those available parking places. The smaller this value, the higher the number of vehicles that the mobile agent will need to visit to collect information about available parking places.
- The *data collecting timeout* is the maximum time that the agent will invest in collecting data. If this task takes too long and this timeout is exceeded, the agent will stop collecting data and try to return the results collected so far to the driver. Therefore, the amount of data collected could be smaller than the amount of available parking spaces required by the driver.
- The *warning timeout* represents an amount of time that, if exceeded, will lead the agent to enlarge the search area (100 m in each direction) to try to find more available spaces.

3.2 Influence of the Interurban Vehicle Density

In this experiment, we evaluate how the retrieval of information about available parking spaces is influenced by the density of vehicles in the interurban area. These areas are characterized by long roads (lengths of tens of kilometers), a small number of intersections, and the lack of buildings that block wireless signals. Due to the long length of the road, the most suitable traffic density measurement unit is the *number of vehicles per linear kilometer* (vehicles/km). We vary this value from 2 vehicles/km up to 20 vehicles/km.

Figure 2(a) shows how the time required to obtain the desired information about available parking spaces decreases as the density of vehicles increases. On the other hand, Fig. 2(b) shows the total number of hops performed by the agent. This value is the number of times that the mobile agent moves successfully from one vehicle to another using the wireless connection, and it can be used as a measure of the bandwidth required by the query processing. When the density of vehicles increases, the number of hops also increases, since the mobile agent is constantly looking for more promising vehicles to try to reach sooner the destination area. The growth in the number of hops, however, is not very steep and it stabilizes for about 8 to 10 vehicles/km.



(a) Time to solve the query (b) Number of agent hops

Fig. 2. Influence of the interurban vehicle density

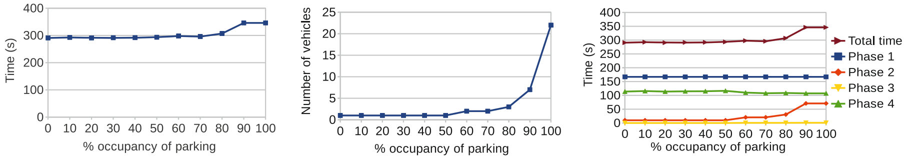
We have also measured the number of vehicles whose data have been processed by the mobile agent looking for information about available parking spaces. This number refers to the number of vehicles that have data about parking places, since there also exist other irrelevant vehicles where the agent can move but that do not contain data about parking spots. Since the query is solved in the phase of data collection (in the urban area) the interurban density has no influence. The number of vehicles processed remains approximately constant, specifically between 6 and 7 vehicles. We omit the figure due to space constraints.

3.3 Influence of the Occupancy of the Parking Spaces

In this experiment, we vary the ratio of available parking places in the scenario, from 0% (all the parking places are available) to 100% (there is no available parking space) in which case the agent will always fail in its task.

Figure 3(a) shows the total time needed to solve the query. The rate of parking occupancy has little impact on the total time, since the mobile agent spends most of the time traveling to the destination area and returning to the originator car, whereas the actual data collection process is performed quite fast.

Figure 3(b) shows the number of vehicles whose data have been processed by the mobile agent. This number remains quite small until the parking occupancy increases considerably. When there is no available parking space (occupancy of 100%), the number of processed vehicles rises abruptly, due to the efforts performed by the agent to try to solve the query looking for non-existing data. In such a case, the timeout established for data collection is reached and the agent returns with an empty response, which might as well be an useful answer.



(a) Query solving time (b) Vehicles queried (c) Time per phase

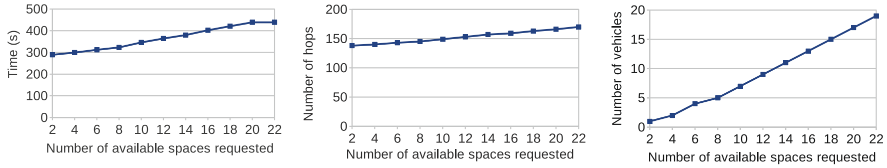
Fig. 3. Influence of the occupancy of the parking spaces

The average time spent in the different phases of the query processing is shown in Fig. 3(c). The first phase (mobile agent traveling to the destination area) takes always the same time, since the ratio of available parking spaces does not have any influence on it. In the second phase (mobile agent collecting data), the time invested increases with the occupancy, since the agent will need to visit more vehicles to find information. Finally, the time invested in the fourth phase (agent traveling back to the origin vehicle) decreases slightly as the occupancy of parking spaces increases; this may seem surprising but, while the mobile agent is collecting data, the origin vehicle keeps traveling towards the target area, so the distance that the agent will need to traverse to reach the vehicle decreases.

We also measured the percentage of the number of parking spaces requested by the user that are collected and returned by the agent. For all the occupancy rates, the result obtained was 100% (i.e., information about all the spaces requested are found by the agent), with the exception of when there are no available parking spaces, where the result was 0%. As we increase the occupancy rate by 10% increments, the following worse case scenario evaluated corresponded to an occupancy of parking spaces of 90%, but with this rate there were still enough available parking spaces to satisfy the query (more specifically, 22).

3.4 Influence of the Number of Requested Available Parking Spaces

In this experiment, we vary the number of available parking spaces that the driver wants to retrieve. The parking occupancy ratio is set to 90%, as this is a quite challenging scenario. The number of available parking spots in the area is initially 22, so we vary the requested parking spaces from 2 to 22, and if the agent cannot find enough available parking spaces within the searching area, it will enlarge it according to the parameter *warning timeout*. Figure 4(a) shows the time needed to solve the query. As expected, the higher the number of parking spots to collect, the higher the time needed by the mobile agent to complete the process. However, the query processing times are not excessive in any case.



(a) Time to solve the query (b) Number of hops (c) Vehicles processed

Fig. 4. Influence of the number of requested available parking spaces

Figure 4(b) shows the number of hops performed by the agent. When the number of requested parking spaces increases, the number of hops also increases, since the agent needs to visit more vehicles. Consistently with this result, Fig. 4(c) shows the number of relevant vehicles that the mobile agent needs to visit, which grows, since the mobile agent needs to visit an increasing number of them to find the information they have about the existing available parking spots.

Finally, the percentage of collected data was 100% in all the cases, with the exception of the case of 22 places, where a 98% of the requested parking spaces (i.e., around 21.5 places, on average) were found despite the efforts performed by the mobile agent, due to the randomness of the information available in the vehicles and the high parking occupancy.

4 Related Work

Most research on query processing in vehicular networks has focused on push-based approaches (e.g., [1]) and only a few works consider pull-based approaches instead (e.g., [3]). The main reason is higher simplicity: push-based approaches avoid some challenges that appear when a query is disseminated in a VANET and the results need to be collected and communicated to the originating vehicle. Moreover, the use of mobile agents for query processing in vehicular networks has not been studied in depth, except for our previous work presented in [10].

Helping drivers to find parking spaces is a topic that has received considerable research attention. So, smart parking systems can be designed and deployed to provide information about parking spaces in specific areas (e.g., see [5]). However, these infrastructure-based solutions are quite expensive and not available globally. On the contrary, it would be interesting to have solutions that are flexible enough to obtain information about any available on-street parking. This motivated the development of proposals that exploit data dissemination in VANETs [4]. Information about the availability of parking spaces can be quite volatile, and some proposals try to take this into account. In [2], an allocation protocol is proposed for sharing information about available parking spaces using only ad hoc communications. This proposal guarantees that the information about a parking space is provided only to a single driver, which avoids competition problems if several drivers receive alerts about the same parking space.

Up to the authors' knowledge, existing proposals to provide information about available parking spaces are push-based, so they cannot be used in scenarios like the one studied in this paper. Moreover, the use of mobile agents to search available parking spaces is also new. Finally, the experimental evaluations focus either on urban scenarios or highways, whereas we have considered a mixed scenario that includes both an urban area and an inter-urban region.

5 Conclusions and Future Work

In this paper, we have presented an approach that uses the technology of mobile agents to find available parking places in a city area by looking *in situ* for that information among the data collected individually by the vehicles that circulate in that area. As opposed to the existing related work, the novelty of our study resides in the use of a pull-based approach to query about parking spaces in areas that are not located near the driver that submits the query (thus allowing new interesting use cases), the use of mobile agents in a parking space searching scenario, and the experimental evaluation with real maps combining both city and interurban areas. The experimental results presented show the performance of the proposal in different conditions and its feasibility.

As a future work, we plan to improve the behavior of the mobile agent to complete the query in less time and with a better use of the available bandwidth.

Acknowledgments. This work has been supported by the projects TIN2016-78011-C4-3-R (AEI/FEDER, UE), TIN2013-46238-C4-4-R, and DGA-FSE.

References

1. Cenerario, N., Delot, T., Ilarri, S.: A content-based dissemination protocol for VANETs: exploiting the encounter probability. IEEE Trans. Intell. Transp. Syst. **12**(3), 771–782 (2011)
2. Delot, T., Ilarri, S., Lecomte, S., Cenerario, N.: Sharing with caution: managing parking spaces in vehicular networks. Mob. Inf. Syst. **9**(1), 69–98 (2013)
3. Delot, T., Mitton, N., Ilarri, S., Hien, T.: GeoVanet: a routing protocol for query processing in vehicular networks. Mob. Inf. Syst. **7**(4), 329–359 (2011)
4. Ilarri, S., Delot, T., Trillo-Lado, R.: A data management perspective on vehicular networks. IEEE Commun. Surv. Tutor. **17**(4), 2420–2460 (2015)

5. Kotb, A.O., Shen, Y.C., Huang, Y.: Smart parking guidance, monitoring and reservations: a review. IEEE Intell. Transp. Syst. Mag. **9**(2), 6–16 (2017)
6. Martinez, F.J., Fogue, M., Toh, C.K., Cano, J.C., Calafate, C.T., Manzoni, P.: Computer simulations of VANETs using realistic city topologies. Wirel. Pers. Commun. **69**(2), 639–663 (2013)
7. Olariu, S., Weigle, M.C.: Vehicular Networks: From Theory to Practice, 1st edn. Chapman & Hall/CRC, Boca Raton (2009)
8. Trillo, R., Ilarri, S., Mena, E.: Comparison and performance evaluation of mobile agent platforms. In: The Third International Conference on Autonomic and Autonomous Systems (ICAS 2007), pp. 41–46. IEEE Computer Society (2007)
9. Urra, O., Ilarri, S.: MAVSIM: testing VANET applications based on mobile agents. In: Vegni, A.M., Agrawal, D.P. (eds.) Cognitive Vehicular Networks, pp. 199–224. CRC Taylor and Francis Group, Boca Raton (2016)
10. Urra, O., Ilarri, S., Trillo-Lado, R.: An approach driven by mobile agents for data management in vehicular networks. Inf. Sci. **381**, 55–77 (2017)
11. Uzcategui, R.A., Sucre, A.J.D., Acosta-Marum, G.: Wave: a tutorial. IEEE Commun. Mag. **47**(5), 126–133 (2009)