Modeling Mobile Agents in Vehicular Networks

Oscar Urra and Sergio Ilarri

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

Abstract. Vehicular networks (VANETs) are mobile ad hoc networks where vehicles that are near each other can exchange data by using wireless communications. Advances in mobile communication technologies have spurred significant research in the exploitation of these types of networks to develop different kinds of data services for drivers. However, these networks are also highly-dynamic and several data management challenges arise to realize their full potential.
We have proposed the use of mobile agent technology for efficient distributed query processing in VANETs. Mobile agents are software entities with the capability to hop among nearby cars in such a way that they can transport themselves to the vehicles storing relevant data. In this paper, using Petri nets, we present a formal model of our approach.

Keywords: Mobile agents, vehicular networks, query processing, data management, Petri nets

1 Introduction

Vehicular networks (VANETs) [3] are mobile ad hoc networks dynamically established among vehicles by using short-range wireless communications and based on communication standards such as WAVE (IEEE 802.11p) [6]. Using these networks, vehicles can exchange relevant data for drivers, such as information about accidents or obstacles on the roads, traffic conditions, available parking spaces, or other moving entities of potential interest. However, several data management challenges appear to fully exploit the potential of VANETs [1]. Most difficulties are due to the fact that the nodes in the network (i.e., the vehicles) are continuously moving, which renders the communication links quite volatile (nodes may appear and disappear at any time and the communication between two distant vehicles is only possible by using multi-hop routing protocols). Thus, for example, two vehicles moving in opposite directions in a highway at high speeds will be within the communication range of each other only during a quite small time window, which constrains the amount of data that can be exchanged.

On the other hand, mobile agents [2, 4] are software entities that have the capability to autonomously move from one computer/device to another during their execution. This technology can bring interesting benefits in distributed systems. Rather than transferring large amounts of data to a node for processing, the code can be moved to the node storing the data for local processing and filtering, thus saving significant network resources. We have previously proposed the use of mobile agents for data management in VANETs [5]. With our approach, a mobile agent can flexibly take autonomous decisions and jump from car to car as needed to reach the target area and query the data sources within that area.

2 Overview of the Data Retrieval Approach

We consider the problem where we need to retrieve data about a certain spatial area, called *interest area* (IA) or target area. We assume that some vehicles within that area can provide the required data. For example, imagine that some vehicles are equipped with sensors of different types (pollution sensors, noise sensors, videocameras, etc.) and our goal is to retrieve environment data about a geographic area, for monitoring or surveillance purposes. In these circumstances, we could try to flexibly exploit the required sensors available in vehicles traveling through those areas. Similarly, we could want to process a query using a distributed approach that access several vehicles in an area to retrieve data from their local databases, such as data exchanged with neighbor vehicles about available parking spaces nearby. Figure 2 shows an overview of the process:

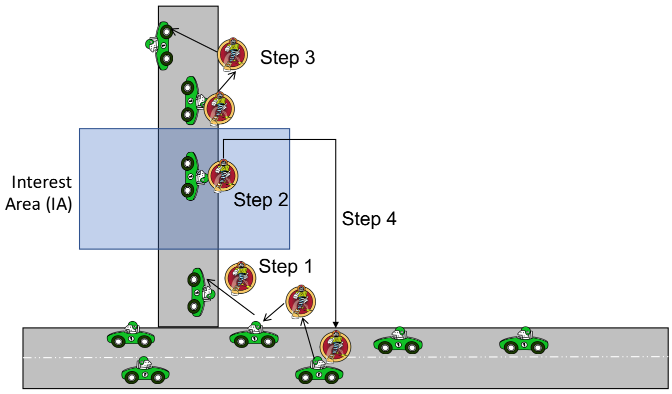

Fig. 1. Overview of the process.

- Step 1: the agent needs to reach the *interest area* (IA).
- Step 2: the agent retrieves data from the vehicles inside the area.
- Step 3: if the vehicle carrying the agent leaves the interest area, the agent will need to find an appropriate strategy to come back.
- Step 4: once the agent has finished the monitoring process, it will need to come back to its origin device and return the results collected.

2.1 Step 1: Traveling to the Interest Area

First of all, the mobile agent needs to reach the interest area. It should be noted that the agent can travel from one place to another by using two complementary mechanisms: by hopping among vehicles (transportation using wireless communications) and by staying in a moving vehicle (transportation via locomotion, using the cars "as taxis").

Figure 2 shows a Petri net that models this stage of the process. The mobile agent is initially created in a given vehicle, as represented by the initial mark in the place "In_Vehicle". Whereas the agent has not succeeded in its attempt to reach the interest area (condition "IA_not_Reached"), it evaluates if there is another vehicle within the communication range that could be a better candidate to transport it to the area. If a better candidate is found (condition "Better_Found"), the agent jumps there, and otherwise it stays in the same vehicle. The process continues until the agent reaches the target area ("IA_Reached"). Notice that there is a transition ("Vehicle_approaching") injecting marks into

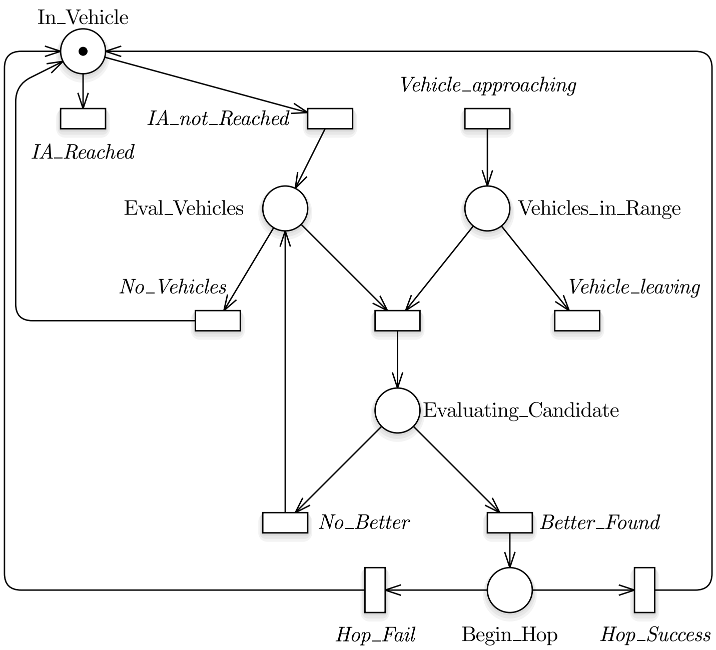

Fig. 2. Model of an agent traveling to the target area.

the place "Vehicles_in_Range" and another transition removing marks from that place ("Vehicle_leaving"), representing the fact that, at any time, new vehicles can start being within (and out of, respectively) the communication range of the vehicle currently transporting the agent. It should be noted that the agent could apply a variety of strategies to decide if a vehicle is a better candidate or not; for example, a simple greedy approach could select as a better candidate any vehicle which is closer to the target area than the current vehicle.

2.2 Step 2: Monitoring the Interest Area

Once the agent is within the interest area, it has to retrieve data by traversing the area and retrieving data from the vehicles located within the area. The agent considers that spatial area as divided into a certain number of spatial cells (of the same size). We assume that the agent needs to visit at least N cells to finish its task (alternatively, we could require visiting a minimum number of cars). Notice that, by increasing or decreasing the number of cells and the value of the parameter N, we could achieve a more fine-grained or coarse-grained monitoring.

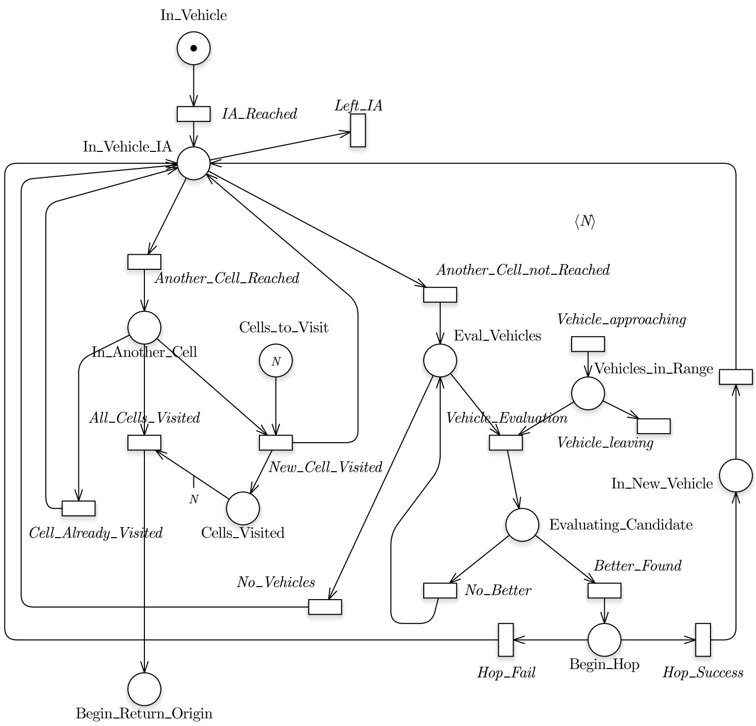

Fig. 3. Visiting the cells within the area: detailed view of the process.

Figure 3 shows a Petri net that models the process of visiting the required spatial cells within the area. Each time that there is an opportunity to visit a cell that has not been previously visited, the agent tries to visit it and, if it succeeds, a mark is injected into the place "Cells_Visited"; if not (transition "Another_Cell_not_Reached"), it will try to reach the cell by traveling to other vehicles if necessary. Once the cell has been visited (transition "Cell_Already_Visited"), the agent will try to visit a different cell (the mark representing the agent returns to "In_Vehicle_IA" and the agent will consider a different cell). When there are no marks left in the place "Cells_to_Visit", which means that there are N marks in the place "Cells_Visited", the transition "All_Cells_Visited" is fired and the agent finishes this stage of the process (a mark representing the agent is put in the place "Begin_Return_Origin") to start the last stage (step 4).

2.3 Step 3: Returning to the Target Area, if needed

Notice that, as shown in Figure 3, the mobile agent can leave the target area if the vehicle that carries it leaves the area (transition "Left_IA"). In that case, the agent will need to temporarily interrupt the monitoring process and find a way to come back, again by jumping from car to car as needed.

Figure 4 shows the process followed by the mobile agent when the car that carries it leaves the interest area.

2.4 Step 4: Returning to the Origin

In the last phase of the process, the mobile agent needs to return to the device/computer that created the agent in the first place, which may be a moving vehicle. For that purpose, we assume that there is an estimation of the location of that device, in such a way that the mobile agent can take jumping decisions to try to reach that location; in case the location is imprecise, the agent will of course need to expand its searching focus to try to reach its "home device". Due to space constraints, we omit the Petri net due to its similarity with Figure 2.

3 Prospective Work

Some advanced aspects of our proposal are not modeled in the Petri nets shown in this paper, such as the potential use of clones (an agent that creates copies of itself) in situations where the reliability or performance of the data retrieval process may be in danger (e.g., scenarios with a low density of vehicles or where the vehicles follow trajectories that do not pass near the target area). Another interesting aspect to consider is the possibility to annotate the Petri nets with performance metrics and probabilities, in order to exploit them for performance evaluation; an important difficulty to achieve this is that we would need to quantify first the impact that different elements in a scenario (density of vehicles, their trajectories, etc.) can have on the performance of the process.

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.

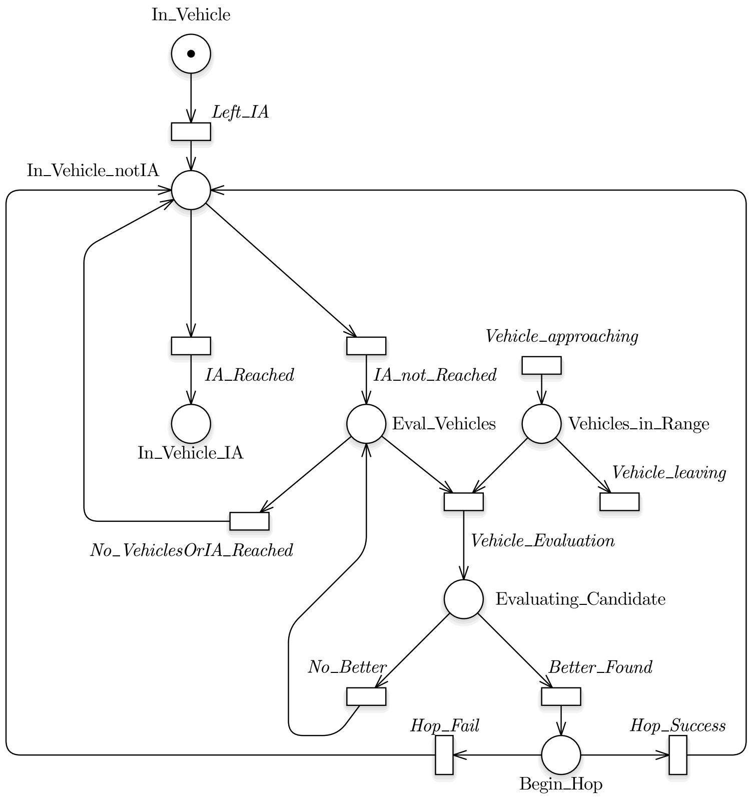

Fig. 4. Returning to the interest area after leaving it.

References

1. Ilarri, S., Delot, T., Trillo-Lado, R.: A data management perspective on vehicular networks. IEEE Communications Surveys and Tutorials, ISSN 1553-877X 17(4), 2420–2460 (Fourthquarter 2015)
2. Milojicic, D., Douglis, F., Wheeler, R.: Mobility: processes, computers, and agents. ACM (1999)
3. Olariu, S., Weigle, M.C.: Vehicular Networks: From Theory to Practice. Chapman & Hall/CRC, first edn. (2009)
4. 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). pp. 41–46. IEEE Computer Society (2007)
5. Urra, O., Ilarri, S., Trillo-Lado, R.: An approach driven by mobile agents for data management in vehicular networks. Information Sciences 381, 55–77 (March 2017)
6. Uzcategui, R.A., Sucre, A.J.D., Acosta-Marum, G.: Wave: A tutorial. IEEE Communications Magazine 47(5), 126–133 (May 2009)