VANET's Mobility Models and Overtaking: An Overview

Djamel Djenouri *, Wassim Soualhi §, Elmalik Nekka §
* CERIST Center of Research, Ben-aknoun, BP 163, Algiers, 16030, Algeria
Email: ddjenouri@mail.cerist.dz
§National Institute of Computer Science (INI), Oued-Smar, Algiers, Algeria.
Email: Mk.wassim@gmail.com, phobosmalik@gmail.com

***Abstract*—The most important feature of Vehicular Ad hoc Networks (VANETs) is the high mobility of nodes, which makes the mobility model one of the most important parameters that should be carefully selected when evaluating any protocol. To correctly and faithfully evaluate protocols in a simulation study, the model should be as realistic as possible. Earlier models used in general mobile ad hoc networks (MANETs), such as the random waypoint, are unsuitable for the VANET application, where the nodes do not move freely in the open area, but on the existing routes that are constrained by many parameters (route intersections, stop and traffic light signals, the presence of other vehicles in front the vehicle, etc.). Some new models taking into account these features have been recently proposed. In this paper we provide an overview of the mobility models newly proposed in literature, which can be used for simulating VANET. Afterwards, we will present our mobility simulator that mimics many of the recent models. Finally, we use this simulator to illustrate the impact of the overtaking on mobility and subsequently on the performance results, a parameter neglected by all the previous simulation studies.**

I. Introduction

In VANET, communications between nodes are direct, i.e. without relying on any dedicated infrastructure, contrary to earlier vehicular networks [1]. Although it is self-organized and easily to deploy, the infrastructureless vehicular network introduces many challenges that should be tackled before real implementations. For instance, to allow communications between nodes (vehicles) which are out of the power range of each other some other intermediaries should act as routers, to remedy the lack of dedicated routers. Thus, a distributed routing protocol needs to be employed. It is mandatory before passing to the real implementation of a routing protocol or any application to evaluate it by simulation. The faithfulness of the simulation results is proportional to the realism of the parameters and models used in the simulation, particularly on the mobility model. Mobility model is the pattern that defines the movements of mobile nodes within the simulated area during a simulation time. The earlier mobility models used in MANET simulation assume the terrain to be without obstacles, and nodes to be able to move freely in the whole rectangular stimulation area. This is realistic for some applications of pedestrians, but it is inappropriate for VANET. More recent studies propose new models constrained to routes and obstacles, and thus are more suitable for VANET. Mobility models proposed for ad hoc networks have been largely studied and surveyed in literature [2], [3]. Nonetheless, the recent models of VANETs have not been surveyed yet. A state-of-the-art material of these solutions is then necessary. In this paper we present the most recent mobility models suitable for simulating vehicular ad hoc networks, as well as a simulator we developed enabling the simulation of many recent models. The rest of the paper is organized as follows: in the next section the related work will be presented, followed by a brief description of the basic mobility models. Section 4 illustrates the VANET's mobility models, followed by the introduction of our simulator and the presentation of the simulation study in section 5. Finally, section 6 concludes the paper and summarizes the perspectives.

II. State-of-the-art

Mobility models can be divided into two categories [2]: entity models and group models, and discuss a variety of these models. They show by simulation how the choice of a mobility model significant effects on the performance results. Except city section [4], all the other models presented in this survey are inappropriate for VANETs. In citeZHR04 the author provide a more recent survey of MANET's mobility models, presenting some ones that consider obstacles in the simulated area. Still, these models are suitable to simulate pedestrian movements, but not vehicles. Some recent research works focus on the vehicular ad hoc network application of MANET. [5] provides a general survey of VANET, and the existing challenges to overcome before the real deployment. In [1] the authors discuss the usefulness of VANET to ensure the vehicular traffic safety and facilities, as well as the advantages it provides compared to the other centralized technologies. Regarding mobility models, some new ones have been specifically proposed for VANET, such as [6], [7], [8]. However, and as far as we know, there is no paper devoted to surveying these new models. In this manuscript, we will present and discuss these novel models, as well as those proposed for the general MANET that applies to VANET. Contrary to earlier models that assume an open area, these ones simulate movements in routes. As we will see, the considered parameters differ from a model to another. For instance, some models use traffic control mechanisms (stop signs or traffic lights) at route intersections, and some others just assume continuous movement at these points. Some assume routes to be single-lane, some others support multi-lanes routes. Some define the security distance, while others just ignore this parameter.

Freeway [9] is a generated-map-based model, in which the simulation area, represented by a generated map, includes many freeways, each side of of which is composed of many lanes. No urban routes, thus no intersections are considered in this model. At the beginning of the simulation, the nodes are randomly placed in the lanes, and move using history-based speeds. A security distance should be maintained between two subsequent vehicles in a lane. If the distance between two vehicles is less than this required minimal distance, the second one decelerates ($a(t)$ is forced to be < 0) and let the forward vehicle moves away. The change of lanes is not allowed in this model. The vehicle moves in the lane it is placed in until reaching the simulation area limit, then it is placed again randomly in another position and repeats the process. This scenario is definitely unrealistic.

Manhattan [9] is another generated-map-based model, introduced to simulate an urban environment. Before starting a simulation, a map containing vertical and horizontal roads is generated. Each of these latter includes two lanes, allowing the motion in the two directions (north/south for the vertical roads and east/west for the horizontal ones). At the beginning of a simulation, vehicles are randomly put on the roads. They then move continuously according to history-based speeds (following the same formula like the freeway model). When reaching a crossroads, the vehicle randomly chooses a direction to follow. That is, continuing straightforward, turning left, or turning right. The probability of each decision are set by the authors respectively to 0.5, 0.25, 0.25. The security distance is also used in this model, and nodes follow the same strategy as in the freeway model to keep this distance. But contrary to the previous model, a vehicle can change a lane at a crossroads. Nonetheless, there is no control mechanism at these points (crossroads), where nodes continue their movements without stoping, which is unrealistic.

City Section Mobility (CSM) [4] can be viewed as a hybrid model between RWP and Manhattan, as it introduces the principle of RWP, especially the pause-time and random selection destination, within a generated-map-based area. At each step of the vehicle's movement a random point is selected from the generated road map, toward which it moves following the shortest path. After reaching that destination, it remains there for a pause-time, then repeats the process. The speed of nodes are constrained by the security distance, along with the maximum speed limit of the road. Rice University Model (RUM) [8] is very similar to CSM, but indeed it uses real maps obtained from the TIGER/Lines database [8]. For each route segment [1], the coordinates are extracted and converted using the Mercator projection [8]. The extracted points are then presented by a graph, where the crossroads are presented by vertices, and routes by weighted arcs. The weight of each arc is dynamically calculated in such a way to mimic the estimated time required for a vehicle to move over the corresponding segment, which is proportional to its maximum authorized speed, its distance, and the number of vehicles it currently contains. Therefore, the lower the weight, more the vehicles move freely in the segment. Note that the maximum authorized speed of a route segment depends on its type. RUM has been compared to RWP, in the evaluation of DSR in two different regions [8]. The first one contains a lot of roads of the same type, with a maximum speed limit set to $56km/h$, while the second one has fewer roads, but of different types (which ranges the maximum speed from $56km/h$ to $120km/h$). The two models engender almost the same results in the first region simulation. However, in the second region simulation (where the mobility is more constrained) the results were clearly different. We mention that like all the previous models, RUM defines no control mechanisms at crossroads.

Stop Sign Model (SSM) [6] is the first model we describe that integrates a traffic control mechanism. In every crossroads, a stop signal is put, which obliges vehicles to slow down and make a pause there. This model is based on real maps of the TIGER/Lines database, but all roads are assigned a single lane in each direction. A vehicle should never overtake its successor (like in all the models presented before), and should tune its speed to keep the security distance. If many vehicles arrive at an intersection at the same time they make a queue, then each one waits for its successor to traverse the crossroads. This results in gathering of nodes, and hugely affects the network connectivity as well as the vehicle mobility (average speeds). According to the authors [6], the problem with this model is the unrealistic disposition of the stop signals, since it is impossible to find a region with stop signals at each intersection. Therefore, they proposed TSM (Traffic Sign Model) [6], in which stop signals are replaced by traffic lights. A vehicle stops at a crossroads when it encounters a red stoplight, otherwise it continues its movement. When the first vehicle reaches the intersection, the light is randomly turned red with probability p (thus turned green with probability $1 - p$). If it turns red it remains so for a random delay (pause-time), forcing the vehicle to stop as well as the ones behind it. After the delay, it turns green then the nodes traverse the crossroads one after the other until the queue is empty. When the next vehicle arrives at the crossroads the process is repeated. TSM and SSM has been evaluated by simulation with ns2 [10]. The results show that the two models are not significantly influenced by the speed of nodes (maximum speeds). This is due to the traffic control models, which slow down the nodes and give more stability to the network [6]. When increasing the pause-time at the intersections, the authors remarked that the performances improved for both models, and that SSM gives better results than TSM when using the same pause-time. The authors argue this by the fact that in SSM nodes always stop at the intersections, unlike TSM. Nevertheless, in reality the pause-time of stop signals is shorter than the one of traffic lights, which makes TSM more stable indeed [6].

STRAW [7] is also a model relying on real maps of TIGER/line. Like the other models (except freeway), the roads include one lane in each direction, and is divided into

[1] a part of the route lying between two crossroads

segments. The model is basically composed of three modules: intra-segment mobility manager, inter-segment mobility manager, and finally the route management and execution module. At the beginning of the simulation, the nodes are placed randomly one behind the other. Then they move using the car following model [7] such that they try to accelerate until reaching the maximum speed of the segment. The first module manages this movement until reaching an intersection. The security distance is maintained, and the overtaking is not allowed. At crossroads the vehicles always slow down, even when they change a segment and turn without a full stop, which is realistic. The second module defines the traffic control mechanism, which includes both stop signals and traffic lights, put on crossroads according to the class of the intersected routes. In addition to this usual control form, the module makes sure that the next segment to take contains enough available space before moving the vehicle toward it. If it is fully busy, the vehicle waits at the crossroads (at the end of the first segment). The last module selects the routes to be taken by each vehicle during the simulation. It implements two approaches: simple straw and straw OD. In the first one, the direction is randomly selected at each intersection. That is, when reaching an intersection, the vehicle randomly decides whether to continue straightforward, or to turn and change the route. On the other hand, in the second approach a destination is selected toward which the vehicle moves using the shortest path. The simulation study made by the authors [7] show that when using STRAW the reception ratio decreases from 43% up to 53% compared to movements in an open area. The results of this simulation also illustrate that the routes arrangement has an impact; scenarios with a high number of crossroads slow down the average speeds of nodes, which improves the reception ratio.

In [11] the authors proposed the so-called MOVE, a VANET's mobility model that uses as compiler SUMO [12], which is a realistic vehicular traffic simulation model. SUMO is an open-source application implemented with java that integrates many realistic parameters, such as realistic accelerations, the usage of real maps reflecting several types of routes (with multiple lanes), as well as traffic lights defining priorities between vehicles. Basically, MOVE is composed of two components; the road map editor and the vehicle movement editor. The former serves to manually and randomly generate a road map, either from TIGER/line files or Google earth files, whereas the latter allows to specify the properties of each vehicle, like the maximum speed, the acceleration, the probability of turning at crossroads, the path to take etc. The information collected by the two editors are sent to the SUMO compiler, then a trace file in ns-2 or Qualnet format is generated. MOVE has been compared by simulation to RWP using AODV. The results show that MOVE causes low reception rate.

The last model we present is proposed by Gorgorin et al [13]. In addition to all the realistic parameters of the previous models, this one implements an overtaking mechanism within multi-lane segments. A vehicle always tries to move on the most right lane (the lowest rang), except in case of overtaking during which it moves left, and intersections in urban environment where it chooses the lane according to the next direction. A hierarchy of vehicle states is defined; free driving, approaching, following, and braking (in the order). When a vehicle is in another state than the free driving, it checks whether higher lanes allow it to pass to a higher state, and thus moves to the left lane to make an overtaking. Identically, a vehicle in a state different from braking checks whether the right lane allows it to at least stay in the same state and then moves right. Moreover, the model allows to specify the driver type, which affect many parameters of the vehicles (speed, acceleration, etc.). Finally, note that the model includes both traffic lights and stop signals at intersections. One of these two different control mechanisms is put at each intersections according to the types of the intersecting segments. The most important parameter added in this model is the overtaking mechanism. However, no study investigating this issue has been done yet.

III. Mobility Simulator Description

The vehicular mobility simulator we developed is multi-platform. In the implementation we used the Microsoft Visual studio .Net environment as C# as the programming language. It allows to simulate movements of vehicles within a selected road map, visualize the movements in real time, and generate mobility trace file to be used by network simulators. The tool provides many visual facilities, and the mobility model implemented is parameterizable in such a way to including many of the models presented in the previous section. Through the simulator visual interface, many parameters related to both the mobility model and the scenario to be simulated are tunable. Hereafter, we illustrate the most important parameters.

The mobility model we implemented can be considered as a mould of models, that can be shaped to simulate several models. This by fixing the following parameters.

- Random method vs shortest path method: It enables the two methods for the generation the route to follow. This allows to define the inter-segment mobility, also known as macro mobility.
- Traffic light: For now, the implemented model includes only traffic lights, no stop signs. The employment of traffic light at intersections could be selected, and the user can define pause time duration when encountering red light for each type of segment intersections. A type of intersection is defined by the combination of the segments forming it. e.g i) intersection of three segments of type 1, two segments of type 1 and one of type 2, ii) three of type 2, etc. The segment type is proportional to the number of lanes it includes in each direction. Our simulator supports the commonest existing types (up to three-lane segments), but it is easily extendable to support four-lane and five-lane segments.
- Pause time after each movement
- Possibility of overtaking: could be either enabled or disabled

This way, many models can be simulated. For instance, when activating shortest path and traffic lights without overtaking we can get a TSM-like model. The RUM model can be fixed by deactivating traffic lights and set pause time to 0. In addition to the previous values, using a grill map allows to simulate Manhattan, and a map without intersections including only multi-lane segments to simulate freeway. But note that models that use stop signs, such as STRAW, are not supported thus far.

In addition to these parameters, our simulator includes some other parameters related to the setting of a scenario, such as:

- Type of the map: The map to be used and its size can be either a real one, picked from Tiger/Line database, or a generic one. For the latter, it can be drawn using a drawing tool included in our simulator, which allows to stack a real map that can be obtained from google-earth and draw the road map to be used over it.
- Number of nodes
- Simulation time
- Maximum speed authorized on each type of segments
- Discreetization time: the time unite that represents the interval between two events (node positions) in the simulator; it is trivial that the less the value, the more precession and realism we get, but this would require more time and resources.

Once the parameters are fixed, the simulation can be launched. The user can choose to visualize the simulation to see the movements of nodes, along with their Ids, speeds, and accelerations in the real time. This is helpful for tuning the value of some parameters, but requires a great deal of time. Once all the parameters are fixed, the user can choose to directly generate the mobility trace file without visualization. The user also has the possibility to specify the format of the output (trace) file. For the time being, the simulator supports GloMoSim and ns2 formats.

IV. Simulation Results

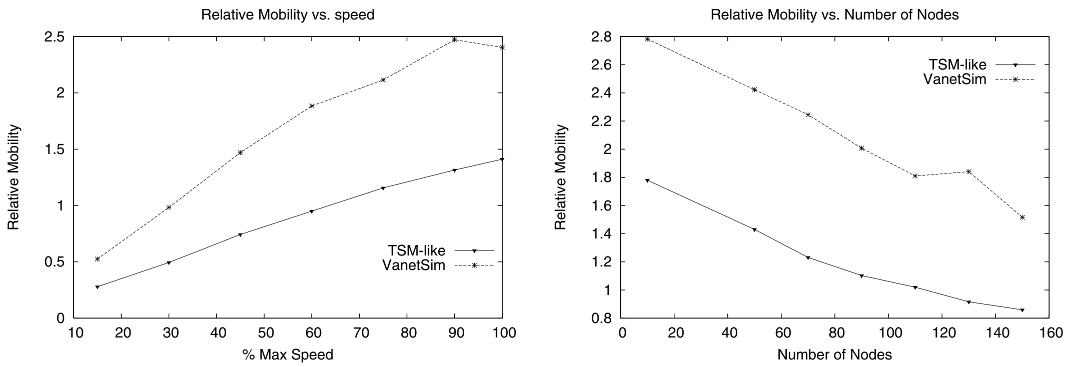
We used our simulator together with GloMoSim to reveal the impact of the overtaking on the mobility. The simulation study consists of comparing two models a TSM-like model and a Gorgorin-like model, that integrates almost all the realistic features, and differers from TSM merely by the fact that it enables the overtaking. We measured the relative mobility, which represents the average changes in distances between nodes, instead of the changes from a fixed terrestrial point. This definition more accurately reflects the network topological, as it considers booth speeds and directions. More details about this metric can found in [14].

Figure 1(a) shows the relative mobility vs. the maximum speed used in the simulation, expressed as the rate of the maximum allowed speed at each segment. We used 130km/hr for 3 lanes segments, 90km/hr for segments of 2 lanes, and 50km/hr for single lanes segment. Hence, the maximum speed a vehicle can reach in our scenario is a ratio of these values, e.g for measurements of 15% the maximum speeds are respectively 19.5km/hr, 13.5km/hr, 7.5 km/hr. The plots of figure 1(a) clearly illustrate the difference in term of relative mobility between a model allowing the overtaking (VanetSim) and the one that does not allow this possibility (TSM-like). The difference between the models increases with the maximum speed, as well as the values of both models. The same metric vs. the number of nodes is depicted in figure 1(b). We remark here that the mobility goes down while the nodes number goes up. This can be argued by the growing in node density. We can also see the huge difference between the two models. The two figures show clearly that the overtaking has a dramatic affect on the mobility. The consequences of this relative mobility on both the packet delivery ratio and the end-to-end delay are clearly illustrated in figures 2(a), 2(b), and figures 3(a), 3(b) respectively. In addition to enhancing the previous results, they show that using a map-based model gives less performances than the open area movement represented by Random Way Point (RWP). Note that for the latest curves, we fixed the max speed to 100%.

V. Summary and Perspectives

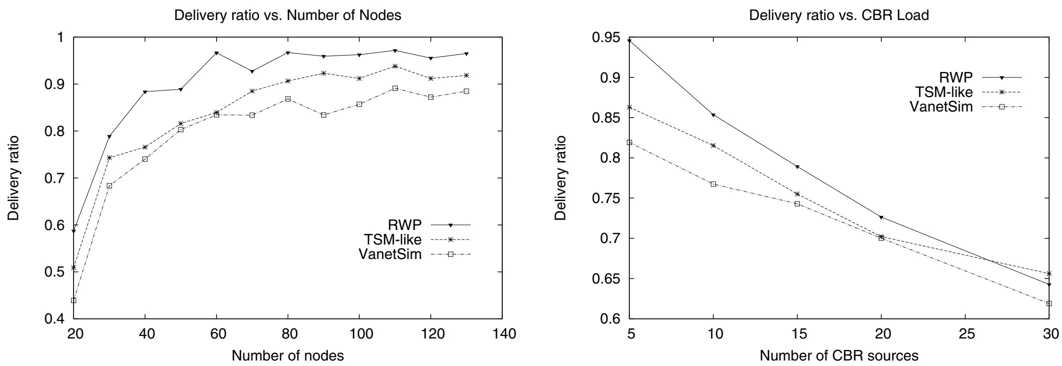
Since the movement of nodes does not happen in an open area but within a route network, earlier mobility models used to simulate general MANETs are inappropriate for simulating VANETs. Map-based models, taking into account the constraints a vehicle faces during its movement, are more realistic and useful for simulating such networks. We dealt in this paper with mobility models that can be used to simulate vehicular ad hoc networks (VANETs), and provided an overview of them. The first three models we described, Freeway, Manhattan, and CSM, are based on unrealistic maps, and do not consider traffic control at intersections. The others are based on real maps, and particularly define control mechanisms at intersections (except RUM). The presence of such intersections where the traffic slows down is obviously realistic in urban areas. This issue has a great impact on the network connectivity and the average speed of nodes. The traffic control of TSM/SSM and STRAW, as well as the one of Gorgorin et al. are the most realistic, as they involves both traffic lights and stop signs, and further considers the place availability when changing a segment. The last model is the only one that considers the overtaking possibility. However, we remarked that the impact of this important parameter has not been investigated in any simulation study.

We developed a mobility simulator that integrates many realistic parameters of vehicular movements, that we used along with GloMoSim to illustrate the effect of overtaking on mobility. The results show that the employment of the overtaking in multi-lane route segments whenever possible has huge affects on the relative mobility. In our perspectives, we plane to make more investigations into the impact of this parameter on the performance parameters, such as the end-to-end delay. Improving our simulator such that to integrate one way routes (for urban environment), and eight lanes (four in each direction) or even more like in some existing highways, as well as the extension to a parallel version and to other



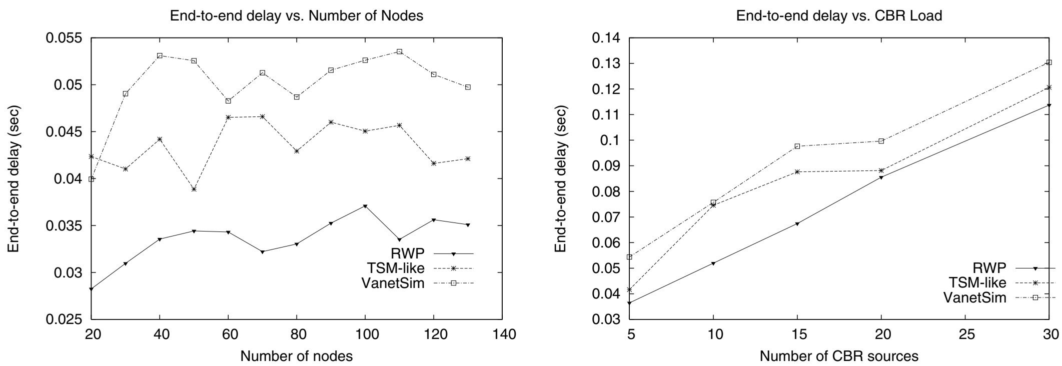
(a) (b)

Fig. 1. Relative Mobility



(a) (b)

Fig. 2. Delivery Ratio



(a) (b)

Fig. 3. End-to-end Delay

mobility trace file formats of other network simulators also represent possible perspectives to this work.

REFERENCES

[1] I. Chisalita and N. Shahmehri, "Vehicular communication: A candidate technology for traffic safety," in *IEEE International Conference on Systems, Man and Cybernetics*, October 2004, pp. 3903–3908.

[2] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, Wiley and sons Publisher*, vol. 2, no. 5, pp. 483–502, 2002.

[3] Q. Zheng, X. Hong, and S. Ray, "Recent advances in mobility modeling for mobile ad hoc network research," in *Proceedings of the 42nd ACM annual Southeast regional conference (ACM-SE)*. ACM Press, 2004, pp. 70–75.

[4] V. Davies, "Evaluating mobility models within an ad hoc network," Colorado School of Mines, Colorado, USA, Tech. Rep. Master's thesis, 2000.

[5] J. Luo and J.-P. Hubaux, "A survey of inter-vehicle communication," School of computer and Communication Sciences, EPEL, Tech. Rep. IC/2004/24, 2004.

[6] A. Mahajan, N. Potnis, K. Gopalan, and A.-I. A. Wang, "Urban mobility models for vanets," in *Proceedings of the 2nd IEEE International Workshop on Next Generation Wireless Networks*, December 2006.

[7] D. R. Choffnes and F. E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks," in *The 2nd ACM international workshop on Vehicular ad hoc networks, VANET'05*. ACM Press, 2005, pp. 69–78.

[8] D. B. J. Amit Kumar Saha, "Modeling mobility for vehicular ad-hoc networks," in *the 1st ACM international workshop on Vehicular ad hoc networks*. ACM Press, 2004, pp. 91–92.

[9] F. Bai, N. Sadagopan, and A. Helmy, "Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks," in *The 22th IEEE Annual Joint Conference on Computer Communications and Networking INFOCOM'03*, 2003, pp. 825–835.

[10] "The Network Simulator: ns2, http://www.isi.edu/nsnam/ns/."

[11] F. K. Karnadi, Z. H. Mo, and K. chan Lan, "Rapid generation of realistic mobility models for vanet," in *IEEE Wireless Communications and Networking Conference*, 2007.

[12] "SUMO Simulation of Urban MObility. http://sumo.sourceforge.net/."

[13] C. Gorgorin, V. Gradinescu, R. Diaconescu, V. Cristea, and L. Ifode, "An integrated vehicular and network simulator for vehicular ad-hoc networks," in *the 20th European Simulation and Modelling Conference*, May 2006.

[14] P. Johansson, T. Larsson, and N. Hedman, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *MOBICOM 99, The Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*. Seattle, Washington, USA, 1999, pp. 15–19.