

Accuracy and Efficiency in Simulating VANETs

Enrique Alba, Sebastián Luna, and Jamal Toutouh

Dept. de Lenguajes y Ciencias de la Computación, University of Málaga,
ETSI Informática, Campus de Teatinos, Málaga - 29071, Spain
{eat,sebastian,toutouh}@lcc.uma.es

Abstract. The evaluation of new communication protocols for *Vehicular Ad-hoc Networks (VANETs)* is a hot topic in research. An efficient design and actual deployment of such software tools is crucial for any VANET. The design phase is difficult and often relies on computer simulation. The later evaluation of protocols in real VANETs is complex due to many difficulties concerning the availability of resources, accurate performance analysis, and reproducible results. Simulation is the most widely solution to make a good design but it presents also an important challenge: the fidelity of the simulation compared to the real results. In this article we measure the differences between the simulation versus the real results with actual moving cars in order to quantify the accuracy of the VANET simulations inside the *European CARLINK Project*¹. After a thorough revision of the state of the art, we here go for an analysis of JANE and VanetMobiSim/ns-2, two simulation frameworks. Later, we have defined the scenario where both, simulations and real tests, will be carried out. Our results show that JANE is more appropriate for simulating applications, while ns-2 is more accurate in dealing with the underlying mobile communication network.

Key words: CARLINK, IEEE 802.11b, Simulation, VANETs, JANE, VanetMobiSim/ns-2

1 Introduction

Vehicular Ad-hoc Networks (VANETs) are created by equipping vehicles with devices capable of wireless communication. The existence of such networks opens the way for a large range of applications: providing real-time information about traffic jams, accidents, and weather; that could be useful for developing a wider set of car vehicles that keeps people connected in metropolitan routes and highways in a clear advance to safer driving, a main issue in today's society.

The evaluation of VANET protocols and applications in outdoor experiments, on using large-scale networks to obtain significant results, is extremely difficult for reasons like the limited/dynamic set of available resources, inaccurate performance analysis, and often irreproducible results. Indeed, it is neither easy nor cheap to have a high number of real vehicles and a real scenario amenable

¹ <http://carlink.lcc.uma.es>

for VANET designers. It is also difficult to analyze protocols performance in a inherently distributed, changing, and complex environment like a VANET [1].

Hence, simulation has become an indispensable tool. It allows to *build* inside a computer a dedicated VANET for the evaluation of protocols: the number of vehicles, the direction and velocity of their movement, the features of the wireless network transceivers, the routing protocol, etc. Simulators also gather statistical data about the network usage during the simulation, which allows to measure the protocols performance. Moreover, it is possible to visualize the VANET in order to easily analyze and conclude on protocol evaluation.

However, due to the complexity of any real scenario in which cars move, a big amount of information related to the signal propagation is missed what is bad news because it plays an important role in the performance of the outdoor experiments: passing by obstacles, reflection problems, coverage signal interferences, etc. Thus, simulation also presents an important drawback: the fidelity of the generated results.

The aim of the CARLINK Project is to develop an intelligent wireless platform among cars. The global scenario considers the cars as data collectors that sends relevant information via wireless technologies up to a central station (where this info is processed). Inside this global scenario, the ad-hoc communications allows the cars to communicate directly with each other without the need of existing infrastructure.

In this paper, we first focus on the simulation of the *CARLINK-UMA scenario*. This scenario consists in two cars moving at 30 km/h while transferring different files between them. The cars follow different mobility models and they are connected through the ad-hoc operation mode of the IEEE 802.11b standard. The goal is to also reproduce the same scenario in outdoor experiments in order to compare the simulated data with the real ones.

We have found that the results of the real tests performed at the University of Málaga (UMA) are an accurate estimation of the data rates that can be achieved when using the ad-hoc WiFi for transferring files between cars. These results could determine the type of applications that could run on top of VANETs. To this goal, we use several simulators, we consider communications and applications, and quantitatively analyze the results.

This paper is structured as follows: Section 2 summarizes the different alternatives for VANETs simulation. Later, VanetMobiSim/ns-2 and JANE are selected as the simulators of interest for the CARLINK Project and Section 3 gives an overview about them. Section 4 defines the scenario where both, the simulations and the real tests, will be carried out. Section 5 presents the simulated versus the real experiments. Afterwards, these results are compared with each other, and finally, Section 6 draws some conclusions about the methodology as well as on the achieved results.

2 Simulation of VANETs

VANETs are a subclass of the Mobile Ad-hoc Networks (MANETs) in which mobility patterns are more complex, since the network topology changes more frequently because of the higher node velocity and the nodes having to fulfil the traffic rules. Therefore, a realistic mobility model implementation is as relevant as a realistic ad-hoc communication network model in order to obtain good quality VANET simulation results. Let us first discuss appropriate tools for simulating either communications and mobility models in next section.

2.1 VANET Simulation Alternatives

Nowadays, we identify different approaches trying to through light on the complex problem of simulating VANETs in a trustworthy manner. First, the most widely used, the desingner could use a traffic simulator for generating realistic vehicular mobility traces that will be used as the input for a mobile ad-hoc network simulator. Second, the designer could use a specially-designed VANET simulator tool. Finally, some MANET application programming frameworks allows the developer to test the applications via simulations.

The first approach used for simulating VANETs consist on using a traffic simulator or a mobility model generator capable of generating mobility traces, which are later evaluated by an existing specific MANET simulator. The public availability of many of these MANET simulators is the main motivation for the success of this approach. However, it has a major drawback: the majority of VANET applications need vehicles to react to network events and it is difficult to be modeled with this scheme of simulation. Most research community adopt ns-2 (network simulator) [8] for MANET simulating. The number of traffic simulators which generates ns-2 format traces is large: the most comprehensive is **VanetMobiSim** [3], however we can also find another as Videlio, RoadSim, CARISMA, VISSIM, and MMTS. There are also traffic simulators that generate traces for other MANET simulators as CORSIM/TSIS, SJ04, SSM/TSM, and STRAW. Finally, TraNS and MOVE combine the SUMO mobility model generator and ns-2 simulator linking them in a unique tool.

The specially-designed VANET simulators join scalable vehicular mobility descriptions and network stack modelling in a single tool. These combined approaches have the big advantage of allowing a direct interaction between the communication network system and the vehicular traffic dynamics, thus, the first can influence the second. However, they also have a major drawback. The level of detail of both modules is necessarily lower than that provided by ad-hoc simulation tools. GGCDI06, MoVES, and the GrooveNet are examples of specific VANET simulators.

Finally, there are some frameworks as JANE [4], a Java-based middleware platform for MANET applications programming. It allows the developer to test the applications in a simulation environment and, also, over real mobile devices. See [5] for more details.

2.2 Selecting a Simulator

Once revised the different approaches for VANET simulation, this section is devoted to giving some recommendations for choosing the most appropriate tool. First of all, the software which is distributed under commercial licenses, as most specific VANET simulators, constitutes a major flaw to adopt them by the research community. Thus, the use of a traffic simulator that consists of the traffic generator traces and the MANET simulator is the most suitable choice. We decided to use ns-2 as MANET simulator, since it is widely used by the research community. So, we need a simulator to generate ns-2 format traces.

Furthermore, the selected simulator has to generate realistic mobility models that reflects as closely as possible the characteristic behavior of the nodes as real vehicles through road traffic by using *macro-mobility* and *micro-mobility* definitions [2]. The simulator should be intuitive with no complex mobility definition. This led us to also experiment with JANE [4], in order to have simulations coupled to applications one of its salient feature.

3 VanetMobiSim/ns-2 and JANE Simulators

According to the previous recommendations, the chosen traffic simulator is VanetMobiSim. It includes several options to specify the roads characteristics (macro-mobility features) and the behavior of the mobility of the nodes (micro-mobility features), the definition of them is done by using intuitive XML code, and the output trace has ns-2 format. Moreover, JANE has been also selected since it allows the development, simulation, and execution of high-level applications in an integrated way.

3.1 VanetMobiSim/ns-2 Simulator

The simulator used for most of the simulations in CARLINK is the combination of the traffic simulator VanetMobiSim and the MANET simulator ns-2 [7].

VanetMobiSim is an extension to CanuMobiSim [9], a generic user mobility simulator. CanuMobiSim provides an efficient and easily extensible mobility architecture, but due to its general purpose nature, it suffers from a reduced level of detail in specific scenarios. VanetMobiSim is therefore aimed at extending the vehicular mobility support of CanuMobiSim to a higher degree of realism. The main characteristics of this simulator are that it is specific for VANETs and an open source platform; it supports both macro-mobility and micro-mobility specification, and it uses intuitive XML code to specify the different simulations. However the most important feature of VanetMobiSim is that it has been validated in actual communication scenarios [3]. Its main drawback is that it offers a poor documentation.

ns-2 [8] is an open source network simulator, so it is freely available and the user is able to modify the source code (C++ and OTcl). This characteristic is really important, since it has allowed us to extend the simulator with the

VDTP protocol [6]. It provides a packet level simulation over a lot of protocols, supporting several transport protocols, several forms of multicast, wired networking, several ad-hoc routing protocols and propagation models, data broadcasting, satellite, etc. It incorporates different traffic generators as web, telnet, CBR (constant bit rate generator), etc. for using them in the simulations. Also, ns-2 has the possibility of using mobile nodes. The mobility of these nodes may be specified either directly in the simulation file or by using a mobility trace file. In our case, the trace file is generated by VanetMobiSim. Finally, other important feature is that it incorporates several add-ons as the visualization tools NAM² (Network Animator) and TraceGraph³.

3.2 JANE: The Java Ad-hoc Network Environment

JANE [4] is an open source Java-based middleware platform which is intended to assist ad-hoc network researchers in application and protocol design. JANE aims at providing a uniform workbench, supporting experiments ranging from pure simulation of mobile devices, over hybrid scenarios with interaction among simulated as well as real life devices, up to dedicated field trials as proof of concepts. Therefore, JANE presents three different execution modes that enable the execution of the tested source code from the simulation to the real devices with a low effort. These execution modes are: *simulation mode*, *hybrid mode*, and *platform mode*. In simulation mode, the complete environment is simulated: the devices, the users and the ad-hoc network. In hybrid mode, the devices and the ad,hoc network are simulated, but real users can interact with the simulation by using emulated graphical interfaces. Finally, in the platform mode the whole setting is real (actual mobile devices as PDAs, cellular phones, etc.).

A development process can be derived from the utilization of these three JANE execution modes. It consists of a cycle which comprises three ordered phases: simulation, emulation, and real execution. It starts with implementing, testing, and evaluating algorithms and applications in a purely simulated environment. In a second step, dedicated mobile devices can be cut out of a simulation run and be transferred to a real mobile device in order to deal with real user interaction and to evaluate the user experience. In a final phase, specific field trials can be defined and executed on real mobile devices.

The main disadvantage of JANE is that this tool is not specialized for VANETs. Therefore, it does not provide realistic mobility models for the simulation of vehicular networks. Nevertheless, its well structured simulation kernel could allow the developer to integrate a more accurate mobility model component for overcoming this weakness.

4 The CARLINK-UMA scenario

The CARLINK Project consider three scenarios for the exploitation of the intelligent wireless platform that is going to be built at the end of the Project [10]:

² <http://www.isi.edu/nsnam/nam>

³ <http://www.angelfire.com/al4/esorkor>

the local weather service, the traffic management service, and the mobile end user service. These services use both, infrastructure and ad-hoc communications, to achieve their objectives. In the CARLINK-UMA scenario we focus on the study of the inter-vehicle communications quality in which the VANETs are based on.

This section describes the scenario where the simulations and the real experiments will be carried out. These conditions will be exactly the same in all the cases in order to make a fair comparison. The goal is to transfer files between two cars connected by using the ad-hoc operation mode of the *IEEE 802.11b MAC Layer Standard* in a line-of-sight scenario. Concretely, each car is equipped with one *PROXIM ORiNOCO PCMCIA transceiver*⁴ connected to a range extender antenna. The wireless network cards output power is 12 dBm and the range extender antennas gain is 7 dBi.

The mobility model consists of a road segment split into two lanes representing bi-directional traffic. Depending on the initial and final positions, we differentiate two scenarios: **Scenario A** and **Scenario B** (see Figure 1). In the first one, both vehicles start at the initial position of the same lane, and they move in the same sense along this lane separated by average 50 m (Figure 1.a). In Scenario B, one vehicle starts the movement at the initial position of the first lane and the other vehicle starts at the final position of the second lane, 500 m separated one from the other, and they move in opposite directions (Figure 1.b). In both cases, each vehicle move with a velocity equal to 30 km/h on average.

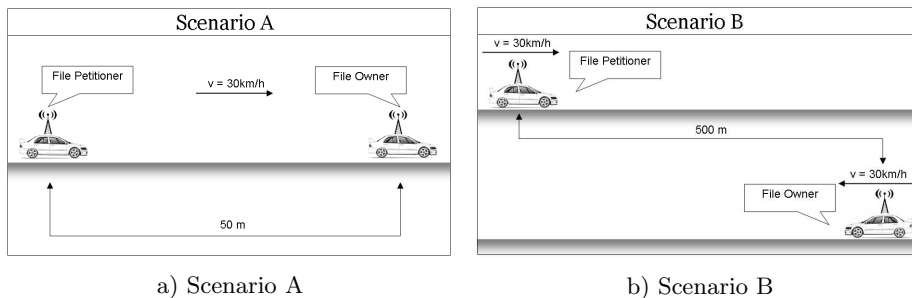


Fig. 1. Mobility models

The experiments were composed of different tests. Each one consisted in continuously transferring a data file in one of the previously specified scenario A or B (Figure 1). We used two different files: **file 1** with 1-MB size (representing traffic notifications, e.g. on road conditions) and **file 2** with 10-MB size (representing multimedia files, e.g. podcasting or streaming to cars).

We use the VDTP protocol [6] to make transfers among the vehicles. For each transfer, VDTP splits the file into several chunks. The chunk size can be configured manually. We have set its value to 25 KB in all the tests.

The complete experiment consisted of **ten repetitions** of every test. The tests were named as follows: **Test A1**, **Test A2**, **Test B1** and **Test B2**. In this notation, the upper case character describes the scenario and the number denotes the file used in each test.

⁴ <http://www.proxim.com>

5 Real Tests versus Simulation

This section presents the differences between real and simulated results. Firstly, we present the results of real tests and simulations for each test. Secondly, we present the difference between each simulator and the real tests.

Figure 2 shows the results of transferring ten times the file type 1 in the Scenario A. The mean transmission time in the real tests is 1.618 seconds, with a mean transmission rate equal to 626.992 KB/s. The mean transmission time achieved using the VanetMobiSim/ns-2 simulations is 1.679 seconds, with an mean data rate equal to 609.778 KB/s. In the case of the JANE simulations the mean transmission time is 1.8 seconds, with a mean data rate equal to 563.812 KB/s. We can notice the high precision of the simulation with ns-2 compared to the actual values showed by the moving cars.

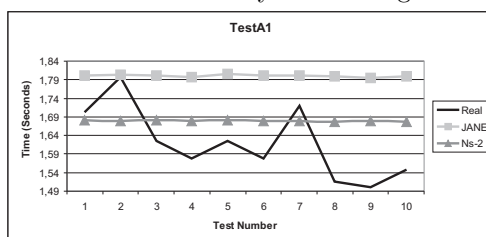


Fig. 2. Individual transmission times for TestA1.

Figure 3 shows the results of transferring ten times the file type 2 in the Scenario A. The mean transmission time in the real tests is 17.328 seconds, with a mean data rate equal to 585.176 KB/s. The mean transmission time achieved using the VanetMobiSim/ns-2 simulations is 16.757 seconds, with a mean data rate equal to 611.053 KB/s, somewhat too optimistic this time. In the case of the JANE simulations, the mean transmission time is 17.9 seconds, with a mean data rate equal to 564.494 KB/s, a better result for this high level tool.

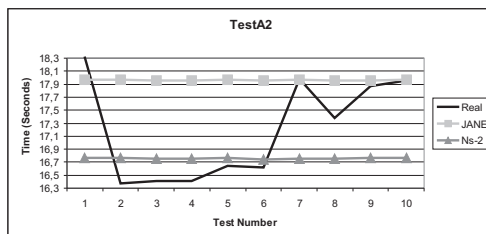


Fig. 3. Individual transmission times for TestA2.

Figure 4 shows the results of transferring ten times the file type 1 in the Scenario B. The mean transmission time in the real tests is 2.732 seconds, with a mean data rate equal to 371.404 KB/s. The mean transmission time achieved using the VanetMobiSim/ns-2 simulations is 2.678 seconds, with an average transmission rate equal to 391.451 KB/s. In the case of the JANE simulations the mean transmission time is 1.8 seconds, with a mean data rate equal to 563.724 KB/s. It seems that small data files get JANE more confused, while ns-2 is specially accurate.

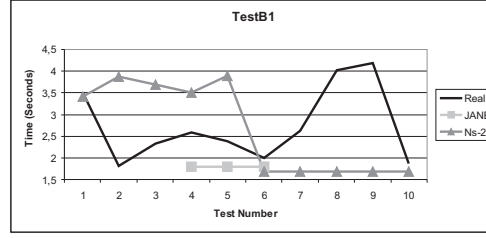


Fig. 4. Individual transmission times for TestB1.

Figure 5 shows the results of transferring ten times the file type 2 in the Scenario B. The mean transmission time in the real tests is 20.198 seconds, with a mean data rate equal to 502.017 KB/s. The mean transmission time achieved using the VanetMobiSim/ns-2 simulations is 19.945 seconds, with a mean data rate equal to 513.397 KB/s. In the case of the JANE simulations, none of these transfers were successful (i.e., none of the files were completely downloaded from the file owner to the file petitioner).

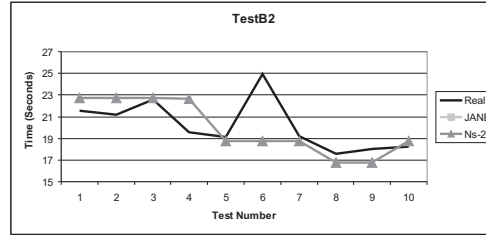


Fig. 5. Individual transmission times for TestB2.

In order to compare all these results, Figure 6 presents the mean data rate for each test. It is easy to check that VanetMobiSim/ns-2 generates more realistic results than JANE. Anyway, let us have a look to the numerical differences presented in Table 1. Each entry (i, j) in this table denotes the absolute difference (in KB/s) between the real experiment and the simulation results with the simulator i in the test j .

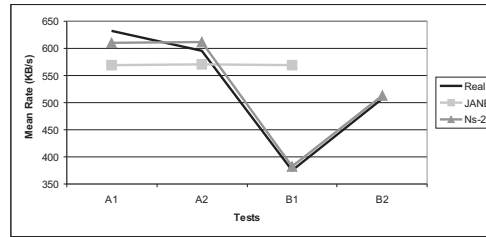


Fig. 6. Mean data rate comparison between real and simulated tests

ns-2 presents the largest difference with the real experiment in Test A2: 25.877 KB/s. JANE presents the largest difference with the real experiment in Test B1: 192.32 KB/s. Moreover, it was not possible to transfer any file completely in JANE with the same conditions as VanetMobiSim/ns-2.

The configuration of the wireless network transceivers for each vehicle were copied from the real specifications of the ORiNOCO card to both simulators (detailed in 4). However, the coverage radius for the cars had very different values in each one: 80 metres in JANE and 100 metres in ns-2. The smaller the coverage diameter the smaller the time frame for the connection between the file petitioner and the file owner in the Scenario B. That could be the reason for the observed differences and advantages of ns-2 since it has a larger coverage. Furthermore, the mean data rate achieved in JANE was also smaller than the one achieved in ns-2 during all the tests. This explains the difficulties for JANE in order to transfer the 10-MB file in the Test B2.

Table 1. Mean data rate differences (absolute value in KB/s) between real and simulation results.

	Test A1	Test A2	Test B1	Test B2
JANE	63.18	20.682	192.32	N/A
VanetMobiSim/ns-2	17.214	25.877	20.04	11.38

6 Conclusions

In this work we have compared simulated versus real experiments about the use of ad-hoc WiFi in VANETs. Firstly, we have deeply study the state of the art in VANET simulation in order to select the most interesting tools for the CARLINK Project: JANE and VanetMobiSim/ns-2. Secondly, we have defined a common scenario for the fair comparison between simulation and real results: the CARLINK-UMA scenario. Finally, we have presented the numerical differences among them.

It is interesting to notice that the times between consecutive file transfers in the simulations are very similar each other, contrary to the times obtained in the real experiments (see figures 2, 3, 4 and 5). The simulation experience shows that the real world communications are quite difficult to simulate in a trustworthy manner. Due to its complexity, a lot of events related to the signal propagation of the wireless transceivers, that play an important role in the real experiments, are missed in the simulations: passing by obstacles, reflection problems, signal interferences, etc. It is advisable to keep this idea in mind when using the simulation results to evaluate any complex scenario before being deployed.

The results presented in Section 5 reveal that VanetMobiSim/ns-2 is the most realistic VANET simulator. Therefore, we have decided to use it in order to perform further complex and larger-scale simulations for the CARLINK consortium. Finally, due to its innovative method for developing new wireless ad-hoc network applications, JANE is useful for testing complex high-level applications deployed on VANETs. Indeed, we have developed two applications that have been successfully tested in real VANETs: FSF and Puzzle-Bubble. These applications are available for download from the CARLINK Project web site⁵.

The results achieved with VanetMobiSim/ns-2 are similar enough to the ones obtained in the real experiments to consider this simulator as a reliable alternative for the evaluation of the communication protocols for CARLINK.

⁵ <http://carlink.lcc.uma.es>

As a future work we plan to perform more complex simulations. Once we have tuned the simulators configuration, we are able to simulate more realistic scenarios in order to predict the performance of the real ones, e.g. urban and highway environments.

The aim of the CARLINK Project is to develop an intelligent wireless platform among cars that will provide three main services to improve the day to day life of European drivers and citizens: *the traffic local weather service* will offer accurate local weather forecast, *the traffic management service* will afford real time traffic information for drivers and *the mobile end user service* gives useful information to the citizens in order to choose the better route to reach their destination, through private or public transportation systems. All these services are supported by wireless communications by means of infrastructure and ad-hoc communications. Therefore, the quality of the inter-vehicular communications in VANETs is crucial for the success of the platform.

Acknowledgments. This work has been partially funded by several institutions: the Spanish Ministry of Industry under contracts FIT-330210-2006-49 and FIT-330225-2007-1 (CARLINK), the Spanish Ministry of Education and Science under contract TIN2005-08818-C04-01 (OPLINK), and the Regional Government of Andalusia under contract P07-TIC-03044 (DIRICOM).

References

1. Fiore, M., Haerri, Filali, F., Bonnet, C.: Understanding vehicular mobility in network simulation. MoVeNet 2007, 1st IEEE international Workshop on Mobile Vehicular Networks, October 2007.
2. Härrri, J., Filali, F., Bonnet, C.: Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy. Technical report, Institut Eurécom (2007)
3. Härrri, J., Filali, F., Bonnet, C., M., Fiore, M.: VanetMobiSim: Generating Realistic Mobility Patterns for VANETs. VANET: '06: Proceedings of the 3rd international workshop on Vehicular ad-hoc networks. Institut Eurécom, France, pp. 96–97. ACM Press (2006)
4. Gorgen, D., Frey, H., Hiedels, C.: JANE-The Java Ad Hoc Network Development Environment. 40th Annual Simulation Symposium (2007)
5. CARLINK::UMA.: D2006/6-Evaluating VANET Simulators for CARLINK Primary Applications. Technical report, University of Málaga (2006)
6. CARLINK::UMA.: D2006/10-VDTP: A File Transfer Protocol for Vehicular Ad hoc Network. Technical report, University of Málaga (2006)
7. CARLINK::UMA.: D1.3.2-VanetMobiSim/Ns-2: A VANET simulator for CARLINK. Technical report, University of Málaga (2007)
8. The Network Simulator, <http://www.isi.edu/nsnam/ns>
9. CANU Project, <http://canu.informatic.uni-stuttgart.de>
10. CARLINK Consortium.: Definition of Scenarios. Technical report, 2007