Vehicular Mobility Simulation with VanetMobiSim*

Jérôme Härri
University of Karlsruhe, Institute of Telematics
76131 Karlsruhe, Germany
haerri@kit.edu

Marco Fiore
Politecnico di Torino, Corso Duca degli Abruzzi 24
10129 Torino, Italy
fiore@tlc.polito.it

Fethi Filali
Christian Bonnet
Institut Eurécom, Dpt. Communications Mobiles
06904 Sophia-Antipolis, France
{filali,bonnet}@eurecom.fr

During the last few years, continuous progresses in wireless communications have opened new research fields in computer networking, aimed at extending data networks connectivity to environments where wired solutions are impracticable. Among these, vehicular communication is attracting growing attention from both academia and industry, owing to the amount and importance of the related applications, ranging from road safety to traffic control and up to mobile entertainment. Vehicular Ad-hoc Networks (VANETs) are self-organized networks built up from moving vehicles, and are part of the broader class of Mobile Ad-hoc Networks (MANETs). Owing to their peculiar characteristics, VANETs require the definition of specific networking techniques, whose feasibility and performance are usually tested by means of simulation. One of the main challenges posed by VANETs simulations is the faithful characterization of vehicular mobility at both the macroscopic and microscopic levels, leading to realistic non-uniform distributions of cars and velocity, and unique connectivity dynamics. However, freely distributed tools which are commonly used for academic studies only consider limited vehicular mobility issues, while they pay little or no attention to vehicular traffic generation and its interaction with its motion constraints counterpart. Such a simplistic approach can easily raise doubts on the confidence of derived VANETs simulation results. In this paper we present VanetMobiSim, a freely available generator of realistic vehicular movement traces for networks simulators. The traces generated by VanetMobiSim are validated first by illustrating how the interaction between featured motion constraints and traffic generator models is able to reproduce typical phenomena of vehicular traffic. Then, the traces are formally validated against those obtained by TSIS-CORSIM, a benchmark traffic simulator in transportation research. This makes VanetMobiSim one of the few vehicular mobility simulator fully validated and freely available to the vehicular networks research community.

Keywords: modeling, simulation, vehicular mobility, validation, VANET, IVC, ITS

SIMULATION, Vol. 00, Issue 0, Xxxxxxxx 2009 000–000

DOI: 10.1177/0037549709345997
Figures 6, 7 appear in color online: http://sim.sagepub.com

---

* This paper is an extended version of a previous publication entitled ‘Vehicular Mobility Simulation for VANETs’ presented at the *40th Annual Simulation Symposium* [1] and includes more details on the VanetMobiSim tool, an extensive discussion on the need for realistic vehicular traffic modeling, a guideline for the development/modeling of realistic vehicular motion patterns and a formal validation of VanetMobiSim against a professional traffic simulator.

1. Introduction

Vehicular communication is regarded as a key technology for improving road safety and comfort through Intelligent Transportation Systems (ITS). The growing interest toward the possible applications of wireless technologies to the vehicular environment has recently led consortia (US IntelliDrive [2], EU C2C-CC [3]) and standardization bodies (IEEE [4], ETSI TC ITS [5], ISO CALM [6]) to develop technologies and protocols for the transmission of data between vehicles and between vehicles and road infrastructures. Vehicular Ad-hoc Networks (VANETs) represent a rapidly emerging, particularly challenging class of Mobile Ad Hoc Networks (MANETs). VANETs are distributed, self-organizing communication networks built up from traveling vehicles, and are thus characterized by very high speeds and limited degrees of freedom in node movement patterns. Such particular features often make standard networking protocols inefficient or unusable in VANETs. When considering the huge impact that the deployment of VANET technologies could have on the automotive market, the growing effort in the development of communication protocols which are specific to vehicular networks is easily explained.

Whereas it is crucial to test and evaluate protocol implementations in real testbed environments, logistic difficulties, economic issues and technology limitations make simulation the preferred method in the validation of networking protocols for VANETs, and a widely adopted first step in the development of real-world technologies. A critical aspect in a simulation study of VANETs is the need for a mobility model which reflects the real behavior of vehicular traffic. It would be desirable for a trustworthy VANETs simulation that vehicular traffic modeling would include both *mobility constraints*, such as movement on streets, obstacles and speed limitations, and realistic *traffic generators*, defining inter-vehicle interaction, intersection handling and overtaking [9].

However, most of the mobility models employed in VANETs simulations ignore some or all of these guidelines, and thus fail to reproduce peculiar aspects of vehicular motions such as car acceleration and deceleration in the presence of nearby vehicles, queuing at road intersections, clustering caused by semaphores, or vehicular congestion and traffic jams. These phenomena in turn generate specific spatial and temporal distributions of vehicles altering network connectivity and routing.

In this paper, we introduce VanetMobiSim [7], a freely distributed and open-source vehicular mobility generator based on the CanuMobiSim architecture [8] and designed for integration with telecommunication network simulators. VanetMobiSim can produce detailed vehicular movement traces employing different motion constraints or traffic generator models and taking into account the interaction of the two, and can simulate different traffic conditions through fully customizable scenarios. We validate the mobility patterns generated by VanetMobiSim by recreating distinctive vehicular mobility effects, such as speed decay with increasing car density, non-uniform distribution of vehicles in urban areas, and shock waves due to stop-and-go perturbations. We formally validate VanetMobiSim by comparing this vehicular traces with those generated by a benchmark traffic simulator in the transportation community. With VanetMobiSim, we provide to the research community working on vehicular networks a freely available and modular tool that is able to generate realistic traces for various network simulators and with which the performance of protocols designed for vehicular communications could be better evaluated.

The rest of the paper is organized as follows. Section 2 motivates the need for realistic motion patterns in vehicular networks, while Section 3 describes a concept map and various guidelines for the development of realistic vehicular mobility models. A detailed description of VanetMobiSim is given in Section 4, while Section 5 presents validation tests on movement traces produced by VanetMobiSim in specific scenarios and by comparison with the benchmark simulator TSIS-CORSIM. Section 6 discusses related work in the field of vehicular mobility modeling for network simulation, and we finally conclude in Section 7.

2. The Need for Realism in Vehicular Traffic Modeling

It has only been in recent times that the networking community has started paying attention to the impact that realistic mobility modeling has on vehicular communications. The use of simplistic mobility models that has characterized most of the literature on topics of mobile and vehicular networks appears as an evident flaw, when considering that vehicular traffic theory has undergone 50 years of accurate studies. When comparing mobility models employed in recent works on vehicular networks and analytical descriptions following well-known approaches of vehicular traffic flow theory, the difference in terms of results is dramatic, and it is clear that such a discrepancy cannot have a null impact on the performance of networking protocols and techniques.

Since the 1960s, vehicular traffic flow theory has introduced models of car driver behavior whose level of realism has been assessed through standard tests. As an example, a minimal requirement for a mobility model is to be capable of recreating the lambda-shaped relation between vehicular flow and density [10]. Low-complexity traffic stream models meet this requirement, even if they look at vehicular mobility as a hydrodynamic phenomenon, and thus do not model the behavior of each car individually. In Figure 1, we depict the aforementioned lambda-shaped relation, as well as the curve relating the speed and out-flow of vehicles, when using the Fluid Traffic Model (FTM) [41] implemented in VanetMobiSim and described in detail later in this paper. Given a straight road, the rea-

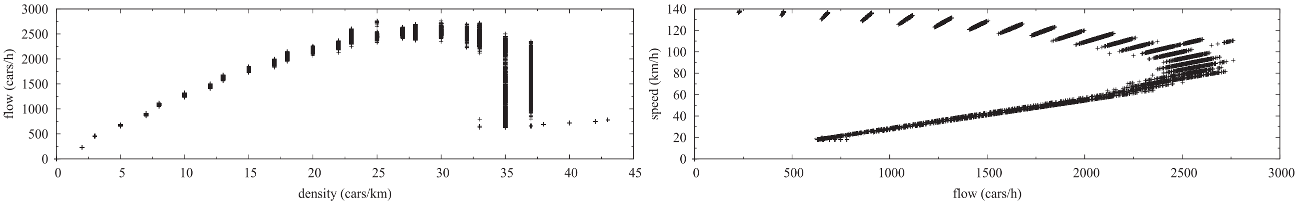


Figure 1. Flow versus density and speed versus flow under the Fluid Traffic Model.

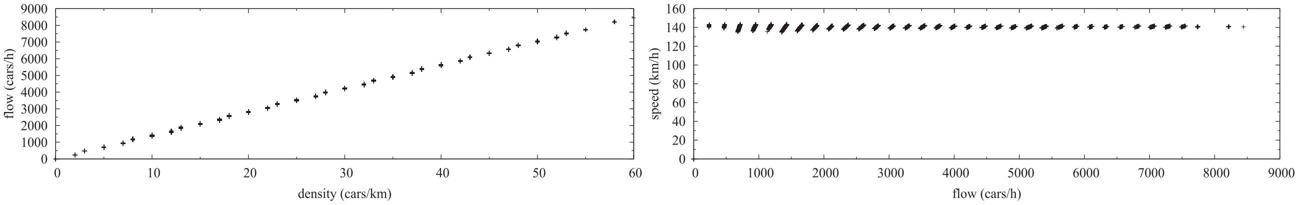


Figure 2. Flow versus density and speed versus flow under the Manhattan model.

soning at the basis of the phenomena is that, as the in-flow rate and consequently the car density is increased, the out-flow of vehicles grows linearly. However, when the critical vehicular density is reached, the road capacity can no longer sustain the arrival rate, leading to queuing phenomena that slow the system down as the density increases further [10].

We performed the same test with the Manhattan model [22], a vehicular mobility representation commonly employed for vehicular network simulation, described by the following set of rules:

$$\begin{aligned}
&v_i(t+\Delta t) = v_i(t) + \eta a \Delta t;\\
&\text{IF } v_i(t) < v_{\min}, \text{ THEN } v_i(t) = v_{\min};\\
&\text{IF } v_i(t) > v_{\max}, \text{ THEN } v_i(t) = v_{\max};\\
&\text{IF } \Delta x_i(t) \leq D, \text{ THEN } v_i(t) = v_{i+1}(t) - a/2;
\end{aligned}$$

where η is a random variable uniformly distributed in $-1, 1$. The results depicted in Figure 2 do not match expectations: even if the Manhattan model implements some bounded randomness in the velocity update, and imposes speed limitations to avoid overlapping of vehicles, the lack of a desired speed and of accurate car following rules make the description unrealistic as the growth in the inflow is producing a linear increase on the car density.

Speed waves represent another condition of vehicular traffic commonly reproduced during the validation process of a mobility model in traffic theory works. These perturbations are known to be generated by heavy traffic conditions on highways or by periodic obstacles such as traffic lights or entering ramps, and are due to the finite response time of drivers to slowdowns determined by such events [11]. As depicted in Figure 3, where slow-speed dark waves move against the direction of traffic in time, a car following a model such as the Intelligent Driver Model (IDM), implemented in VanetMobiSim and discussed later in the paper, can correctly recreate this phenomenon. The equivalent plot obtained using the Manhattan model appears as a white image, since all of the vehicles maintain the maximum speed, and is thus not shown here. On the other hand, in the plot on the right of Figure 3, the FTM fails to reproduce the desired behavior in that case, since this model does not include a car-to-car interaction description.

Another typical proof of the validity of a vehicular mobility model is its response to dynamic situations, such as that occurring in a queue of cars in the presence of an obstacle ahead that is suddenly removed. In that case, it is expected that the model forces the drivers to slow down while approaching the obstacle and then to accelerate again once the impediment is removed. This is actually what we can observe in Figure 4, when the IDM is used. Each line represents the evolution of speed over time of one car and we plot curves for the first 20 vehicles in the queue. It can be noticed that the first vehicle slows down as the obstacle closes in, and that the cars behind follow the leader's speed dynamics with some delay due to the drivers' reaction time. When the obstacle is removed, just before the leading car stops completely, the vehicles start accelerating again toward full speed. Cars back in the queue experience a different speed evolution, as they are far from the obstacle and are thus still moving at high speed when the impediment is removed. The same is not true when the Manhattan model is used, as the model prevents vehicle overlapping by abruptly reducing to zero the speed of the leading vehicle when it reaches the obstacle. Furthermore, it is not able to induce a free-flow acceleration due to the lack of a desired speed description.

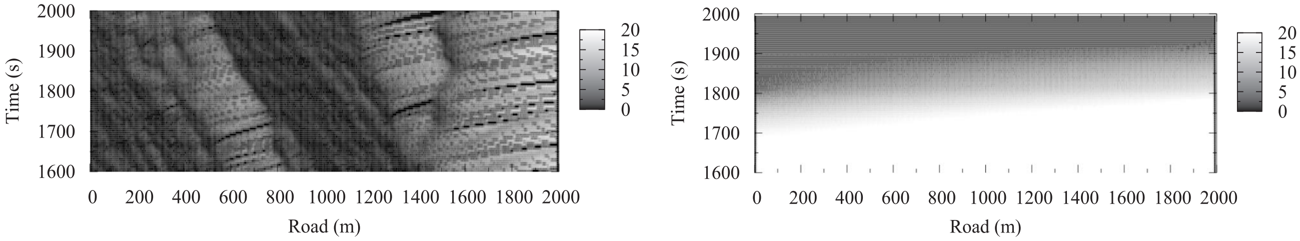


Figure 3. Speed versus time and space in a highway scenario, in presence of increasing car in-flow, when using the Intelligent Driver Model (left) and the Fluid Traffic Model (right). Speed is expressed in meters per second.

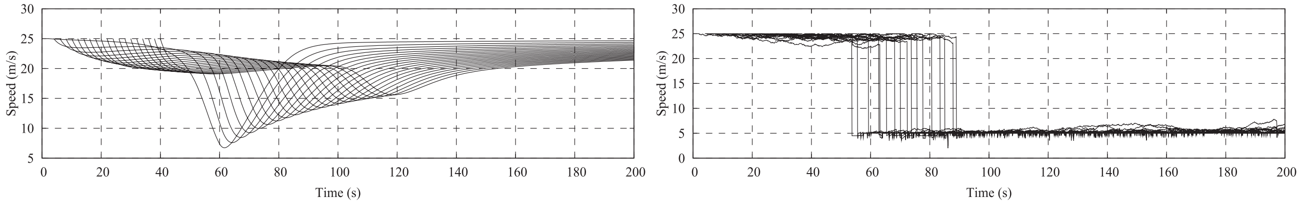


Figure 4. Evolution of speed for the first 20 vehicles belonging to a queue of cars meeting an obstacle which is then suddenly removed. The plots refer to the case in which the Intelligent Driver Model is employed (left) and that in which the Manhattan model is used (right).

The cars in the queue are forced to strictly follow the leading vehicle behavior, and thus describe similar curves. The result is shown in the plot on the right of Figure 4.

As evidenced by the results shown in this section, mobility models can perform very differently when facing traffic theory tests which challenge their realism. Unfortunately, models commonly employed nowadays for vehicular networks simulation often fail even the most basic tests. In this paper, we therefore plan to develop a tool that conforms to traffic theory validation tests and that is freely available to the vehicular networking community.

3. Guidelines for the Development of Realistic Vehicular Motion Patterns

During our work, we followed the concept map proposed in [9], which defines a generic framework for vehicular mobility classification and identifies key features that should be included in a vehicular mobility simulator in order to obtain realistic motion patterns. In this section, we recall and improve the guidelines of [9] for the development of realistic vehicular mobility models.

In the literature, vehicular mobility models are usually classified as either macroscopic or microscopic [26]. The *macroscopic* description models gross quantities of interest, such as density or mean velocity of cars, treating vehicular traffic according to fluid dynamics, while the *microscopic* descriptions consider each vehicle as a distinct entity, modeling its behavior in a more precise but computationally more expensive way.

However, a micro–macro approach provides more of a broad classification than a formal description. A more precise way that we suggest for looking at mobility models is to identify two functional blocks: motion constraints and traffic generator. *Motion constraints* describe the relative degree of freedom of each vehicle. Macroscopically, motion constraints are streets or buildings, but microscopically, constraints are modeled by neighboring cars, pedestrians, or by diversities either due to the type of car or to the driver's habits. The *traffic generator*, on the other hand, defines different kinds of cars, and deals with their interactions according to the environment under study. Macroscopically, it models traffic densities, speeds and flows, while microscopically it deals with properties such as inter-distances between cars, acceleration, braking, and overtaking. Also important in realistic motion modeling are *time patterns*, which can be seen as a third functional block and describe different mobility configurations for a specific hour of the day or day of the week.

According to the concept map in Figure 5, mobility models intended to generate realistic vehicular motion patterns should include the following features.

- *Accurate and realistic topological maps:* street topologies should manage different densities of roads, should contain multiple lanes, different categories of streets and associated speed limitations.
- *Obstacles:* obstacles should be intended in a broad sense, as both constraints to car mobility and hurdles to wireless communications.
- *Attraction/repulsion points:* initial and final destinations of road trips are not random. Most of the time, many drivers are driving toward similar

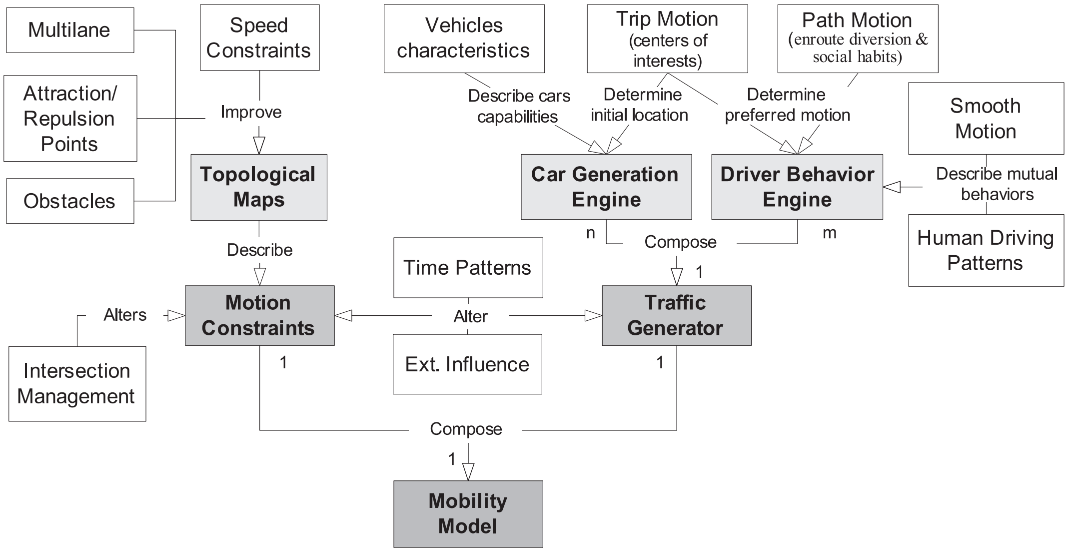


Figure 5. Concept map of for the design of vehicular mobility models.

final destinations or attraction points (e.g. office), or from similar initial locations or repulsion points (e.g. home), typically creating bottlenecks.

- *Vehicles characteristics:* each category of vehicle has its own characteristics, which has an impact on a set of traffic parameters. For example, macroscopically speaking, some urban streets and highways are prohibited to trucks depending on the time of the day. Microscopically speaking, acceleration, deceleration, and speed capabilities of cars and trucks are different. The accounting of these characteristics alters the traffic generator engine when modeling realistic vehicular motion.
- *Trip motion:* a trip is macroscopically seen as a set of source and destination points in the urban area. Different drivers may have diverse interests which affect their trip selection.
- *Path motion:* a path is macroscopically seen as the set of road segments taken by a car on its trip between an initial and a destination point. As may also be observed in real life, drivers do not randomly choose the next heading when reaching an intersection as is the case in most vehicular networking traffic simulations. Instead, they choose their paths according to a set of constraints such as speed limitations, time of the day, road congestion, distance, and even the driver's own habits.
- *Smooth deceleration and acceleration:* vehicles do not abruptly break and move; deceleration and acceleration models should be considered.
- *Human driving patterns:* drivers interact with their environments, not only with respect to static obstacles but also to dynamic obstacles, such as neighboring cars and pedestrians. Accordingly, the mobility model should control vehicles mutual interactions such as overtaking, traffic jams, or preferred paths.
- *Intersection management:* this corresponds to the process of controlling an intersection and may either be modeled as a static obstacle (stop signs), a conditional obstacle (yield sign), or a time-dependent obstacle (traffic lights). It is a key part in this framework that however only has an influence on the *motion constraint* block, as the *traffic generator* block cannot not see the difference between a stop sign or high-density traffic. Both are interpreted as a motion constraint.
- *Time patterns:* traffic density is not identical during the day. A heterogeneous traffic density is always observed at peak times, such as rush hours or during special events.
- *External influence:* some motion patterns cannot be proactively configured by vehicular mobility models as they are externally influenced. This category models the impact of accidents, temporary road works or real-time knowledge of the traffic status on the motion constraints and the traffic generator blocks. Communication systems are the primary source of information about these external influences.

Although it is a promising approach, the proposed guidelines suffer from non-negligible limitations. Indeed, parameters defining the different major classes such as *topological maps*, *car generation engine*, or *driver behavior engine* cannot be chosen randomly but must reflect realistic configurations. Therefore, owing to the large complexity of such a project, the research community took more simplistic assumptions and neglected some blocks. For example, most models available nowadays include a topological map or at least a graph as a motion constraint. However, they do not include speed constraints or more generally attraction or repulsion points. The *car generation engine* block is widely absent from all models, and the *driver behavior engine* is limited to smooth accelerations or decelerations. Our objective is to develop and present a traffic generator that is compliant with the proposed framework and implements most of the features in the concept map of Figure 5.

4. VanetMobiSim

VanetMobiSim is an extension to CanuMobiSim [8], a generic *user mobility* simulator. CanuMobiSim is a platform- and simulator-independent software coded in *SUN Java* [12] and produces mobility traces for the following network simulators: *ns-2* [31], *GloMoSim* [32], and *QualNet* [33]. It is able to integrate user-defined or Geographic Data File (GDF) map [37] topologies, contains a variety of mobility models and provides an easily extensible mobility architecture. CanuMobiSim, however, suffers from a limited level of detail, which makes unsuitable for the modeling of vehicular mobility.

VanetMobiSim therefore aims at extending the vehicular mobility support of CanuMobiSim to a higher degree of realism. By extending CanuMobiSim, VanetMobiSim notably inherits all of its features but also contains the following novel features: integration of *TIGER maps* [36] and *Voronoi topologies*, a complete *road topology characterization*, *intersection* modeling, *overtaking* capabilities, *traffic light* management, the *IDM-IM*, the *IDM-LC*, and the *MOBIL* mobility models. VanetMobiSim also differs from CanuMobiSim by its tighter structure around a *spatial model*[1]. We show in this paper that VanetMobiSim's original features are crucial for a realistic vehicular mobility modeling. In this section, we outline the structure and characteristics of VanetMobiSim and provide details of the resulting vehicular mobility support.

4.1 Software Architecture

Similarly to CanuMobiSim, VanetMobiSim is a modular discrete event simulator based on SUN Java. The software architecture of VanetMobiSim is articulated around two extension objects: the *Universe* and the *Node*, the former modeling static objects, while the latter models movable objects. As may be observed in Figures 6(a) and 6(b), extension objects contain extension modules the role of which is to model the motion constraints and the traffic generator blocks described previously. Conceptually speaking, extension objects represent the actors of a simulation, while the extension modules represent the actors' particular desired details or behaviors. In the remainder of this paper, extension objects and modules will irrespectively be referred to as 'modules'. The *Universe* module is considered in VanetMobiSim to have a God's view as it contains references to all nodes and to the full spatial environment. For the simulation, all modules are attached to the central coordinator and are activated when an event requires actions (see Figure 6(c)). Each feature contained in VanetMobiSim is implemented as a module and is loaded at start-up from an .xml scenario file.

As illustrated in Figures 6(a) and 6(b), each module must implement three key methods:

- *load*, this method is called while loading the scenario in order to feed all required parameters to the module;
- *initialize*, this method is called when VanetMobiSim starts in order to initialize the module;
- *act*, when the module requires an action, this method is called, for example when a car changes its speed or direction.

Owing to this modular structure, adding new features to VanetMobiSim only requires adding a new module, completing its three methods and loading it in the .xml scenario file. At that time, it is not possible to dynamically load new modules or change their parameters during simulation. VanetMobiSim is currently being extended to add this feature through a user-friendly configuration and visualizing Graphical User Interface (GUI).

4.2 Data Structure

In order to have an efficient data structure, VanetMobiSim as CanuMobiSim, uses the GDF data structure. It is based on three levels of detail in the geographic objects to which is attached the description of their attributes and their relationships. We briefly describe here the VanetMobiSim data structure. For more details, we refer to the description of GDF [37].

4.2.1 Features

A feature is the description of a real-world element such as a street, building, car, or intersection. In order to provide a gradual level of detail, three levels of precision are available. The relations between the various levels are described by *Relationships*.

---

[1] Unlike CanuMobiSim, all modules in VanetMobiSim are connected (either strongly or loosely) to a spatial model as this module is in charge of modeling the motion constraints required by the traffic generator.

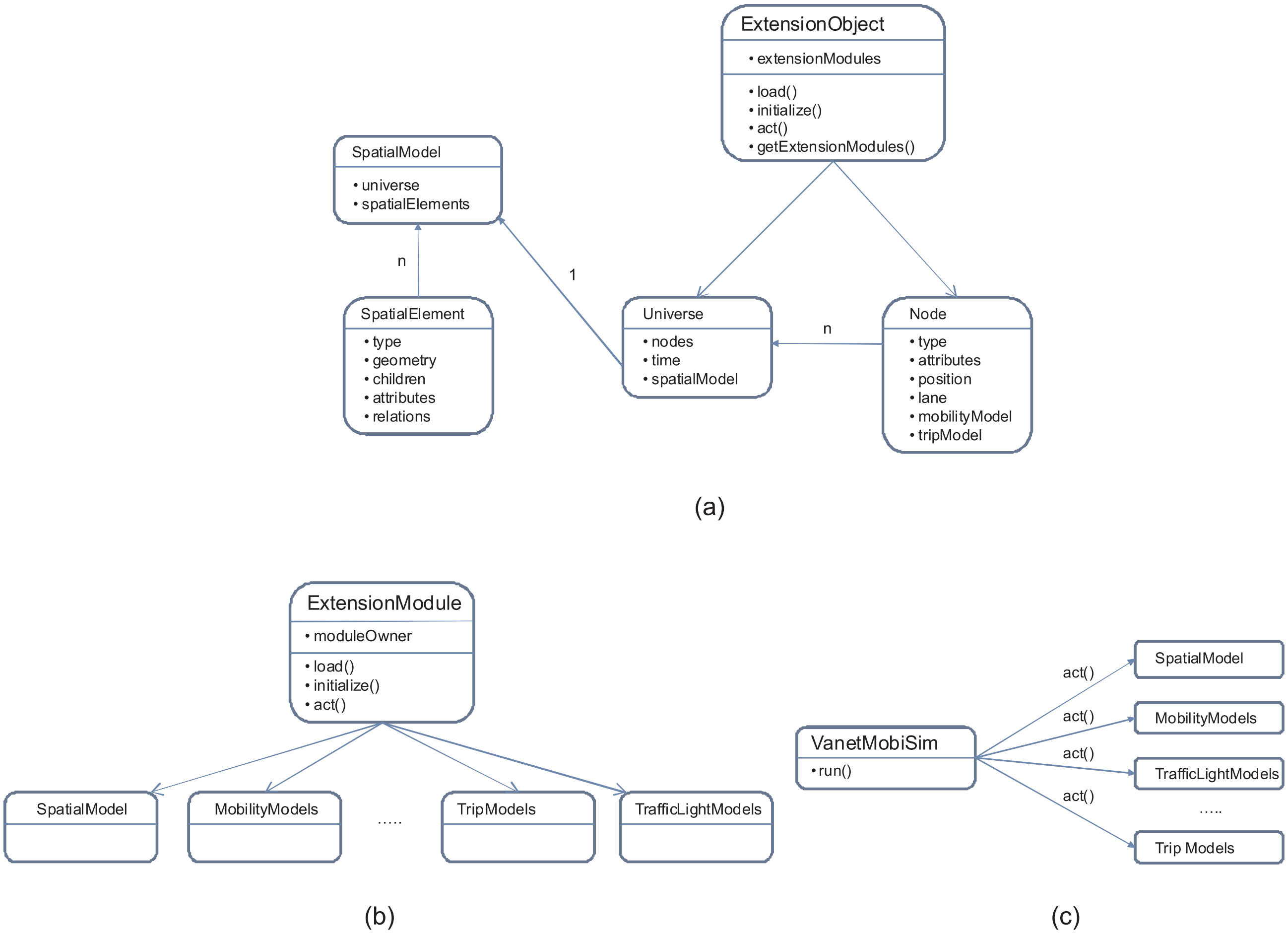


Figure 6. VanetMobiSim software architecture: (a) extension object and spatial model concept; (b) extension module concept; (c) discrete event calls.

- *Level 0*: This represents the geometrical layer containing the vertices and edges describing a higher layer feature.

- *Level 1*: This represents simple, mostly atomic features such as a *Road Furniture*, a *Road Element*, a *Junction* or a *Car*.

- *Level 2*: This level contains complex features regrouping Level 1 features such as a *Street* containing multiple *Road Elements*.

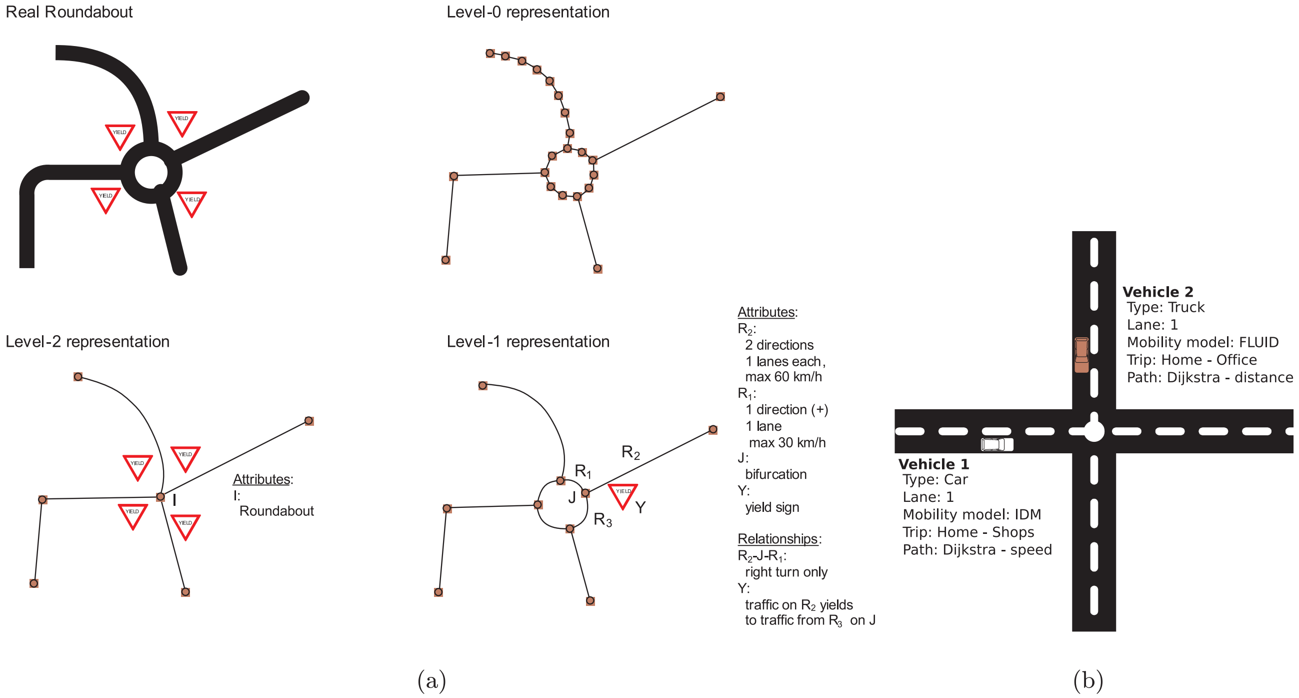
4.2.2 Relationship

In order to describe the interactions between features, a *relationship* description is employed. For example, *Is Capital of* is the relationship between *Paris* and *France*. In VanetMobiSim, relationships are critical as they inter-link the different features such as road elements and junctions. Relationships are therefore crucial to path and trip planning, and notably to intersection management.

Figure 7(a) shows a typical illustration of this approach. When a car reaches the roundabout at junction *J* in Figure 7(a), it needs to know which direction it is allowed to take. The relationship provides the turning restrictions and priorities such that the car knows it can only turn right and must yield. Note that such relationship can also include attributes such that it could only relate to a specific class of vehicle.

4.2.3 Attributes

The properties of real-world objects are represented as attributes. In GDF and therefore in VanetMobiSim, attributes are classified according to attribute types, each one representing a well-defined property of a real-world object. For example, *Speed Limit* is an attribute of a *Road Element*. Regardless of the description layer, each feature or relationship may contain attributes.



(a) (b)

Figure 7. GDF-like VanetMobiSim data structure: (a) motion constraints data structure; (b) traffic generator data structure.

In Figure 7(a), we can see that the road elements R_2 and R_1 have different speed limit attributes due to the different curvatures. Furthermore, the road furniture element Y has the $yield$ attribute as it is a *Yield Sign*.

Motion constraints are not the only data structure inspired from GDF. The Traffic generator is also based on it. As depicted in Figure 7(b), a vehicle may contain several attributes and parameters, such as its type (car, truck, etc.), position and speed. More interestingly, each vehicle has a specific micro-mobility, trip, and path model such that we can model individual vehicles or a group of them differently.

Considering the parameters from the VanetMobiSim data structure, the software structure of each spatial model element is illustrated in Figure 6(a).

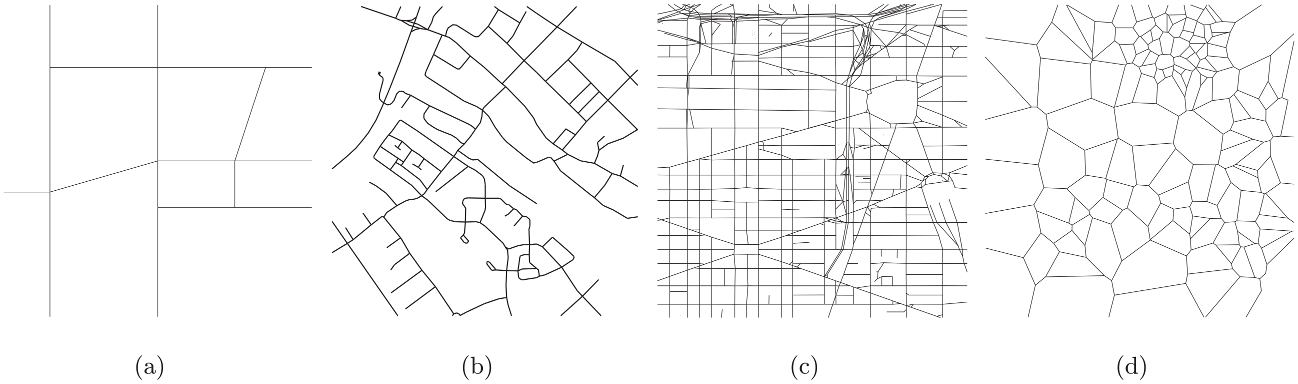
4.3 Motion Constraints Modeling

As illustrated in Figure 5, motion constraints do not only take into account the road topology, but also the road structure (unidirectional or bidirectional, single- or multi-lane), the road characteristics (speed limits, vehicle-class-based restrictions) and the presence of traffic signs (stop signs, traffic lights, etc.). We emphasize that all motion constraints are loaded in the VanetMobiSim's *Spatial Model* and can be used irrespectively of the traffic generator. All of these different aspects of the macro-mobility are discussed in detail in the remainder of this section.

4.3.1 Road Topology Definition

The selection of the road topology is a key factor for obtaining realistic results when simulating vehicular movements. Indeed, the length of the streets, the frequency of intersections, or the density of buildings can greatly affect important mobility metrics such as the minimum, maximum, and average speed of cars, or their density over the simulated map. VanetMobiSim allows the definition of the road topology in the following ways:

- *User-defined graph*: the road topology is specified by listing the vertices of the graph and their interconnecting edges.
- *GDF map*: the road topology is imported from a GDF [37]. Unfortunately, most GDF file libraries are not freely accessible.
- *TIGER map*: the road topology is extracted from a map obtained from the TIGER database [36]. The level of detail of the maps in the TIGER database is not as high as that provided by the GDF standard, but this database is open and contains digital descriptions of wide urban and rural areas of all districts of the United States. In fact, topology descriptions from the TIGER database are becoming quite common in VANETs simulations.
- *Clustered Voronoi graph*: the road topology is randomly generated by creating a Voronoi tessellation



(a) (b) (c) (d)

Figure 8. Road topologies examples: (a) user-defined topology; (b) GDF map topology; (c) TIGER map topology; (d) clustered Voronoi.

on a set of non-uniformly distributed points. This approach is similar to that proposed in [38], but we also consider the presence of areas with different road densities, which we refer to as *clusters*. The number of clusters and their density are customizable to represent diverse geographical characterizations in the same map, such as city centers, suburban areas, or the countryside. The clustered Voronoi graph can be particularly useful to rapidly generate large road topologies.

In all of these cases, the road topology is implemented as a graph over whose edges the movement of vehicles is constrained. The first two models are part of the original CanuMobiSim tool, while the latter two are introduced by VanetMobiSim. Examples of different VanetMobiSim topologies are illustrated in Figure 8.

4.3.2 Road Topology Characterization

As stated before, the concept of modeling vehicular motion constraints is not limited to movement limitations deriving from graph-based mobility, but also includes all aspects related to the road structure characterization, such as directional traffic flows or multiple lanes, speed constraints, or intersection crossing rules. None of these aspects is present in CanuMobiSim, thus the following enhancements are introduced by VanetMobiSim:

- Introduction of roads with multiple lanes in each direction.
- Physical separation of opposite traffic flows on each road.
- Definition of independent speed limits on each road of the topology.
- Implementation of traffic signs at each road intersection. By default, intersections are fully regulated by stop signs. Alternatively, it is possible to regulate traffic at intersections by means of traffic lights.

Note that, for the road topology characterization to have an impact on vehicular mobility, a strong interaction between the motion constraints description and the traffic generator models that define drivers behavior is required. Thus, the traffic generator must be designed to take road characteristics into consideration. This issue is discussed in Section 4.4, when presenting mobility models which account for the presence of traffic signs at intersections and multi-lane streets.

4.4 Traffic Generator Modeling

The traffic generator includes all aspects related to an individual car's behavior, from the selection of target movement destinations and routes to reach them, to speed and acceleration modeling. The Traffic generator description plays the main role in the realism of car movements, as it is responsible for effects such as smooth speed variation, cars queues, traffic jams and overtakings.

4.4.1 Vehicular Movement Pattern Selection

Vehicular traffic schemes in urban scenarios are far from being random. Indeed, cars tend to move between points of interests, which are often common to many drivers and can change in time (e.g. offices may be strong attraction points, but mainly during the first part of the morning). Accordingly, VanetMobiSim exploits CanuMobiSim capability of building movement patterns up from the cooperation of a *trip generation module*, which defines the sets of points of interest, and a *path computation module*, whose task is to compute the best path between those points.

Two choices are given for the trip generation module. The first is a *random trip*, as the start and stop points of movement patterns are randomly selected among the vertices of the graph representing the road topology. The second is an *activity sequences* generation, in which a set of start and stop points are explicitly provided in the road topology description, and cars are forced to move among them. In particular, multiple sets of points of interest can be specified, along with the probability matrix of a vehicle switching from one set to another.

Independently from the trip generation method employed, the path computation, i.e. the selection of the best sequence of edges to reach the selected destination, can be performed in three ways. The first method selects the shortest path to destination, running a *Dijkstra's algorithm* with edges cost inversely proportional to their length. The second method does not only consider the length of the path, but also the *traffic congestion level* by weighting the cost of traversing an edge also on the number of cars traveling on it, thus modeling the real-world tendency of drivers to avoid crowded paths. The last method, which is not present in the original CanuMobiSim, extends the other two by also accounting for the *road speed limit* when calculating the cost of an edge, in a way that fastest routes are preferred.

The combination of trip generation and path computation methods offers a wide range of possibilities, when the definition of vehicular movement paths is a factor of interest in mobility simulation.

4.4.2 Microscopic Vehicular Mobility

Three broad classes of microscopic models, featuring an increasing degree of detail, can be identified depending on whether the individual speed of vehicles is computed (i) in a deterministic way, (ii) as a function of nearby vehicles behavior in a single-lane scenario, or (iii) as a function of nearby vehicles behavior in a multi-flow interaction (i.e. urban) scenario.

CanuMobiSim provides implementations for models belonging to the first two classes. The *Graph-Based Mobility Model* (GBMM) [39], the *Constant Speed Motion* (CSM) [8] and the *Smooth Motion Model* (SMM) [40] fall into the first category, as the speed of each vehicle is determined on the basis of the local state of each car and any external effect is ignored. They all constrain a random movement of nodes on a graph, possibly including pauses at intersections (CSM) or smooth speed changes when reaching or leaving a destination (SMM). The movement is random in a sense that vehicles select one destination and move towards it with random constant speed and along a shortest-length path, ignoring (and thus possibly overlapping with) other vehicles during the motion. While these models may work for isolated cars, they fail to reproduce realistic movements of groups of vehicles.

The FTM [41] and IDM [42] are instead part of the second class, as they account for the presence of nearby vehicles when calculating the speed of a car. These models describe car mobility on single lanes, and do not consider the case where multiple vehicular flows have to interact, as in presence of intersections.

The FTM describes the speed as a monotonically decreasing function of the vehicular density, forcing a lower bound on speed when the traffic congestion reaches a critical state according to the following equation

$$s = \max\left[s_{\min}, s_{\max}\left(1 - \frac{k}{k_{\text{jam}}}\right)\right]$$

where s is the output speed, $s_{\min}$ and $s_{\max}$ are the minimum and maximum speed, respectively, k_{jam} is the vehicular density for which a traffic jam is detected, and k is the current vehicular density of the road the respective node is moving on. This last parameter is given by $k = n/l$, where n is the number of cars on the road and l is the length of the road segment itself. According to this model, cars traveling on very crowded and/or very short streets are forced to slow down, possibly to the minimum speed, if the vehicular density is found to be higher than or equal to the traffic jam density. On the other hand, as less-congested and/or longer roads are encountered, the speed of cars is increased towards the maximum speed value. Thus, the FTM describes traffic congestion scenarios, but still cannot recreate queuing situations, nor can it correctly manage the behavior of cars in the presence of road intersections. Moreover, no acceleration is considered and it can happen that a very fast vehicle enters a short/congested edge, suddenly changing its speed to a very low value, which is definitely a very unrealistic situation. Finally, the implementation of the FTM in CanuMobiSim cannot model the zero speed case, as the condition $s = 0$ causes cars to stop and no longer move, since a loop is entered, in which the vehicular density remains constant in time if all vehicles are still and in turn vehicles cannot increase their speed if the vehicular density does not decrease. It is thus necessary that $s_{\min} > 0$.

On the other hand, the IDM characterizes drivers behavior depending on their immediately preceding vehicle, thus falling into the so-called *car-following* models category. The instantaneous acceleration of a vehicle is computed according to the following equations

$$\frac{dv}{dt} = a\left[1 - \left(\frac{v}{v_0}\right)^4 - \left(\frac{s^*}{s}\right)^2\right]$$

and

$$s^* = s_0 + \left(vT + \frac{v\,\Delta v}{2\sqrt{ab}}\right)$$

In the first equation, v is the current speed of the vehicle, v_0 is the desired velocity, s is the distance from preceding vehicle and s^* is the so called *desired dynamical distance*. This last parameter is computed as shown in the second equation, and is a function of the minimum bumper-to-bumper distance s_0, the minimum safe time headway T, the speed difference with respect to front vehicle velocity Δv, and the maximum acceleration and deceleration values a and b. When combined, these formulas give the instantaneous acceleration of the car, divided into a 'desired' acceleration $1-(v/v_0)^4$ on a free road, and braking decelerations induced by the preceding vehicle $(s^*/s)^2$.

VanetMobiSim adds two original microscopic mobility models, both of which account for the interaction of multiple converging flows by acting consistently with the road infrastructure, and thus fall into the third category mentioned above. These models extend the IDM description, which is the most realistic among those present in CanuMobiSim, in order to include the management of intersections regulated by traffic signs and of roads with multiple lanes. We underline the need for a strong interaction between motion constraints road characterization and traffic generator microscopic mobility that arises from the following paragraphs. We also would like to emphasize that, as both models extend IDM, they are also able to reproduce a lambda-shaped relation between vehicular flow and density, or any traffic theory validation test.

The first new micro-mobility model is referred to as *Intelligent Driver Model with Intersection Management* (IDM-IM). It adds intersection handling capabilities to the behavior of vehicles driven by the IDM. In particular, IDM-IM models two different intersection scenarios: a crossroad regulated by stop signs, or a road junction ruled by traffic lights (implemented by the motion constraints description as described in Section 4.3.2). In both cases, IDM-IM only acts on the first vehicle on each road, as IDM automatically adapts the behavior of cars following the leading car. Every time a vehicle finds no intermediate car between itself and an intersection regulated by stop signs, the following parameters are used by IDM-IM

$$\begin{cases} s = \sigma - S \\ \Delta v = v \end{cases}$$

where σ is the current distance to the intersection and S is a safety margin, accounting for the gap between the center of the intersection and the point the car would actually stop at. Thus, compared with the IDM, the distance from the preceding vehicle is substituted by the distance to the point the vehicle has to stop at. On the other hand, the speed difference is set to the current speed of the car v, so that the stop sign is seen as a still obstacle. This allows vehicles to freely accelerate when far from the next intersection, and then to smoothly decelerate as they approach a stop sign. Once a car is halted at a stop sign, it is informed by the motion constraints block of the number of cars already waiting to cross the intersection from any of the incoming roads. If there are no other cars, the vehicle may pass. Otherwise, it has to wait for its turn in a first-arrived–first-passed and right-hand rule policy. The current version of this *all-stop* intersection management therefore only allows one vehicle passing the intersection at a time. However, more realistic managements considering the trajectories followed by each vehicle in the intersection could be also added. Moreover, it could also be envisioned to model the popular *yield* sign notably used in roundabouts[2], therefore allowing vehicles to move through a junction without stopping if no other vehicle is crossing their trajectories. Both aspects are left to future extensions.

When a vehicle is heading toward a traffic light intersection, it is informed by the motion constraints block about the state of the semaphore. If the color is green, passage is granted and the car maintains its current speed through the intersection. If the color is red, crossing is denied and the car is forced to decelerate and stop at the road junction, using the modified IDM parameters as in the case for a stop sign.

It may also be stressed out that vehicles behavior can dynamically vary in the presence of traffic lights, according to red-to-green and green-to-red switches. The former case is illustrated in Figure 9. In the configuration represented by solid-line curves, a vehicle starts moving at $t = 0$ s, accelerates up to the desired speed, decelerates as the traffic light becomes closer, and eventually comes to a full stop in front of the traffic light. The movement only starts over again when the traffic light turns green at $t = 110$ s. This can be easily observed in both figures. In a second configuration represented by dashed-line curves in Figure 9, a vehicle starts its movement at $t = 35$ s and thus arrives in proximity of the traffic light at about $t = 110$ s, i.e. right on time to observe the traffic light switching to green. Since the vehicle is still in its deceleration phase and has not yet halted, it accelerates again as shown by the upper figure. Thus, in the second configuration, vehicles do not stop at the intersection. Therefore, as shown in the upper plot, the dashed-line curve is always greater than zero, while in the lower image, the advantage in terms of speed experienced by the vehicle in the second scenario leads to an increased traveled space.

In the case of a green-to-red switch, a minimum breaking distance $\bar{s}$ is evaluated by means of simple kinematic formulas as

$$\begin{aligned} \bar{s} &= vt - \frac{\kappa b}{2}t^2 \\ &= v\left(\frac{v}{\kappa b}\right) - \frac{\kappa b}{2}\left(\frac{v}{\kappa b}\right)^2 \\ &= \frac{v^2}{2\kappa b} \end{aligned}$$

which describes the space needed to come to a full stop as a function of the current speed of the vehicle v, the time t and the deceleration value κb. The last parameter represents the maximum safe deceleration, i.e. the IDM comfortable braking value b scaled by a factor $\kappa \geq 1$. The final expression above is obtained by substitution of t

---

[2] The modeling of a roundabout as a spatial model is supported by VanetMobiSim as illustrated in Figure 7(a).

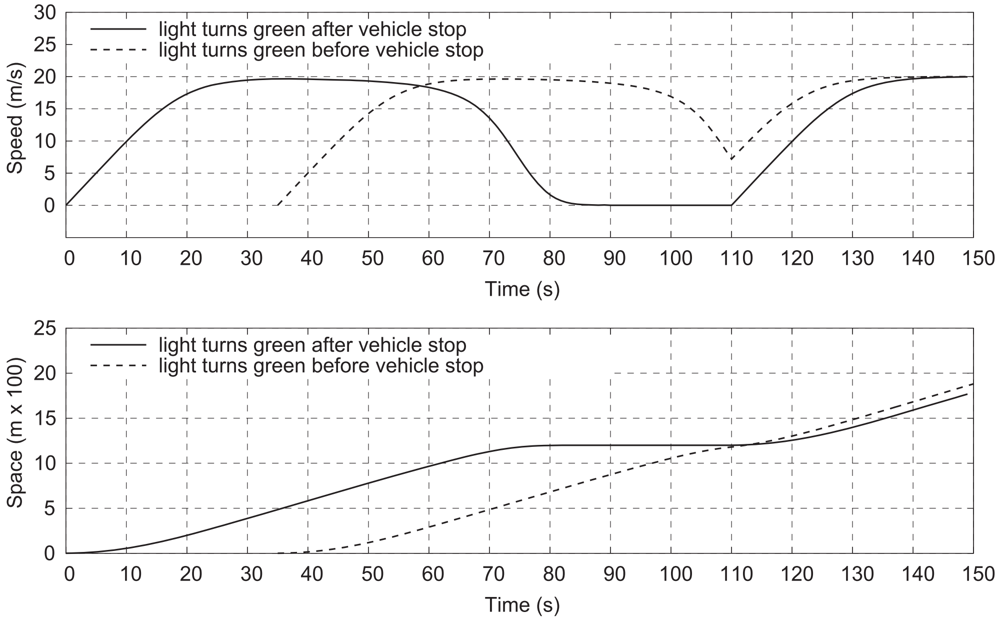


Figure 9. Traffic light *red-to-green* scenario. A vehicle, driven by the IDM-IM setup in Table 3, starts its movement from zero speed, and travels towards a red traffic light. The upper figure shows the evolution of speed in time, while the lower figure depicts the car movement on the road versus time (the upper curves can be seen as the time derivative of the lower curve).

with $(v/\kappa b)$, which is the time at which a zero velocity is reached by inducing a constant deceleration κb on current speed v. Upon computation of $\bar{s}$, if the vehicle finds that it is not possible to stop before the intersection, even by braking as hard as possible, i.e. if $\bar{s} > \sigma - S$, then it crosses the intersection at its current speed. Otherwise, it stops by applying a strong enough deceleration. This reproduces a real-world situation, since drivers only stop if safety braking conditions can be respected when a traffic light switches to red. Examples of driving behaviors in the presence of a green-to-red semaphore are shown in Figure 10. Different curves represent different movement start times, i.e. different positions of the vehicle under study with respect to the traffic light when it switches from green to red (40, 100, 200 and 400 m, respectively). The 40 m case, represented by solid-line curves, is an example of the lack of safety conditions, since 45 m $= \bar{s} > \sigma - S = 40$ m. The car is too close to the traffic light when the color changes, thus the vehicle maintains its speed and does not stop. In the other cases, the safety condition is satisfied, and the vehicle comes to a complete stop in front of the semaphore (as shown in the lower figure). However, the deceleration starts at various distances from the traffic light, leaving different reaction margins to the driver. As proved by the upper plot, this results to a peculiar braking evolution with more comfortable decelerations as the distance from the semaphore increases when the color switches to red.

The second model we introduce is named *Intelligent Driver Model with Lane Changes* (IDM-LC). It extends the IDM-IM model with the possibility for vehicles to change lane and overtake each others by taking advantage of the multi-lane capability of the macro-mobility description detailed in Section 4.3.2. Two issues are raised by the introduction of multiple lanes: the first is the separation of traffic flows on different lanes of the same road, while the second is the overtaking model itself.

As far as the first problem is concerned, vehicular flows on parallel lanes of the same road are separated by forcing the car following model to only consider vehicles traveling in the same lane. However, as the number of lanes can vary from one road to another, a vehicle approaching a crossroad will receive from the motion constraints block the information about the structure of the road it is going to move to. It can then adopt one of the following behaviors.

- If the lane the vehicle is currently moving in is also present in the next road on its path, then it moves through the intersection and keeps traveling on the same lane in the next street.
- If the lane currently used by the vehicle does not exist in the next road, then it tries to merge to its right as it approaches the junction. If it cannot do this, e.g. because the lane to its right is very crowded, it stops at the intersection and waits until a spot becomes available.

On the overtaking model itself, the MOBIL model [43] is employed, mainly due to its implicit compatibility with the IDM. This model adopts a game theoretical approach

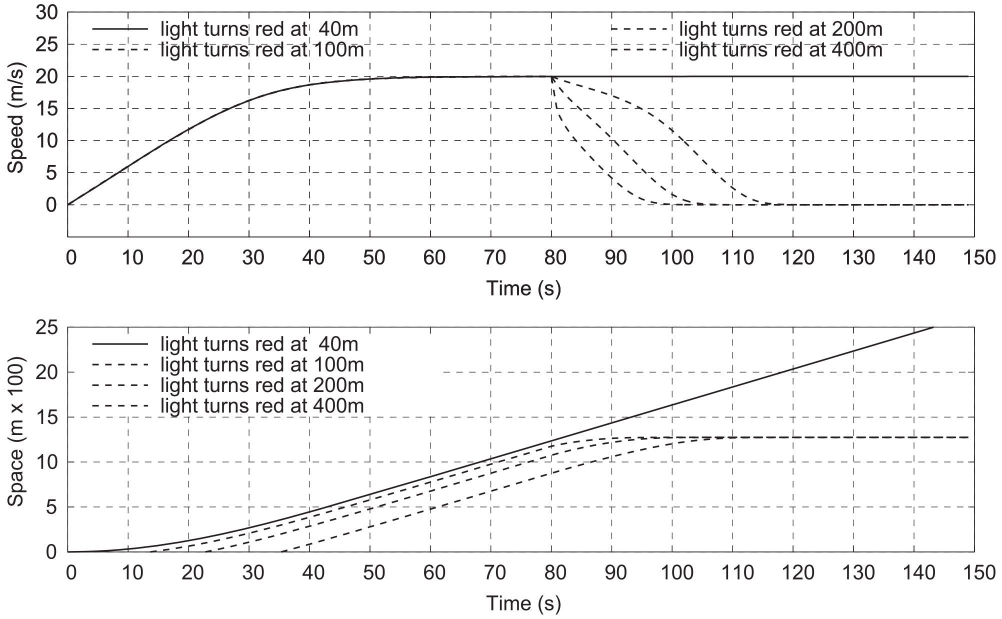


Figure 10. Traffic light *green-to-red* scenario. A vehicle, driven by the IDM-IM setup in Table 3, starts its movement from zero speed, and travels towards a green traffic light, which turns into red at time $t = 80$ s. The upper figure shows the evolution of speed in time, while the lower figure depicts the car movement on the road versus time (the upper curves can be seen as the time derivative of the lower curve).

to address the lane changing problem, allowing a vehicle to move to a different lane if the lane change minimizes the vehicles overall braking. Such requirement is fulfilled when the two conditions

$$a^l - a \pm a_{\text{bias}} > p\left(a_{\text{curr}} + a_{\text{new}} - a^l_{\text{curr}} - a^l_{\text{new}}\right) + a_{\text{thr}}$$

and

$$a^l_{\text{new}} > -a_{\text{safe}}$$

are verified. In the first inequality, a is the current acceleration of the vehicle, i.e. dx/dt in the IDM formulas, while a^l is the equivalent acceleration, computed in the case that the vehicle moved to an adjacent lane l. Similarly, a_{curr} and a^l_{curr} describe the acceleration of the car which currently follows the vehicle we are considering in the case the vehicle under study stays on its lane, or in the case it moves to another lane l. Finally, a_{new} and a^l_{new} represent the acceleration of the car which would become the new back vehicle if the car under study changed its lane to l, before and after a possible lane change of the latter. The model allows a vehicle to move to lane l if the first inequality is verified, that is if, in terms of acceleration, the advantage of the driver who changes its lane $a^l - a$, is greater than the disadvantages of the following cars $a_{\text{curr}} - a^l_{\text{curr}}$ and $a_{\text{new}} - a^l_{\text{new}}$. The MOBIL model also considers a politeness factor p, which scales the right-hand term, in a way that, for values of p near (or above) one, a polite behavior towards other drivers is maintained, while, as p moves to (or below) zero, the driver can become selfish or even malicious. The threshold acceleration a_{thr} introduces a minimum acceleration advantage to allow a lane change in order to avoid lane hopping in border cases. The bias term a_{bias} is instead added to favor movements to one side: in our case, this bias value is added to the advantage computed for movements to the right and subtracted for movements to the left, thus reproducing the real-world tendency of drivers to stay on their rightmost lane on a multi-lane road. Finally, in any case, the safety condition expressed by the second equation above must be verified for the lane change to occur, meaning that the vehicle in the back does not have to brake too hard (its deceleration must be over the safe value a_{safe}) as a consequence of the lane change.

4.5 VanetMobiSim Simulation Performance

A critical point for any simulator lies in its performance in terms of speed. We evaluate the capabilities of VanetMobiSim by measuring the time it requires to complete a simulation under realistic settings, when the number of vehicles increases.

More precisely, the simulation environment we consider for this test is as follows.

- *Motion Constraints*: 2,000 × 2,000 m TIGER map of Washington, DC, including traffic lights at all intersections.

Table 1. VanetMobiSim performance evaluation.

Number of cars	Simulation length (s)
50	11
100	17
300	43
500	73
1,000	157
2,000	393
5,000	2,008

- *Traffic Generator*: Activity-based trip modeling with Dijkstra shortest path and the IDM-IM micro-mobility model.
- *Evaluation Platform:* Intel Core 2 Duo, 2.2 GHz with 2 GB RAM. The Java platform is SUN Java 1.6 [12].

We configure VanetMobiSim to run a 500 s simulation with an increasing number of vehicles and record the effective simulation length. Results are illustrated in Table 1. As can be seen, VanetMobiSim can easily simulate a large number of vehicles in a reasonable time: up to 2,000 vehicles can be simulated in less than 400 s. When considering 5,000 vehicles, the simulation ends after approximately 30 minutes: such a value appears relatively low, considering the amount of time required to simulate even an extremely lightweight network protocol with such a number of highly mobile nodes in a network simulator like ns-2 and fed with the traces generated by VanetMobSim. In other words, the bottleneck in the simulation of a large-scale vehicular network would not be the mobility traces generation, but the network stack simulation. Nevertheless, VanetMobiSim is currently being improved in order to further reduce this simulated time.

4.6 Imperfect Driver Modeling

In the previous sections, we described in detail how vehicular modeling is modeled in VanetMobiSim in order to provide realistic mobility patterns. We notably integrated the most advanced traffic flow models for vehicular traffic. However, these models are based on an accident-free modeling where vehicles act such that a dangerous situation leading to an accident never occurs. This approach could raise questions related to the very claim of realism of the generated mobility pattern. Indeed, a driver is far from being perfect and usually applies traffic rules very loosely.

A specific research field called *Driver Behavioral Modeling* recently generated increased attention from the community as a solution to further fit to realistic driver modeling. VanetMobiSim currently does not contain any behavioral or accident-prone model as such, but provides some level of freedom in its configuration in order to model non-standard driver behaviors. For example, the MOBIL overtaking model contains a *politeness* factor that controls how a driver will consider the impact of its maneuver on other drivers, which in turn could model its decision to stay on a specific lane irrespectively to the traffic rules. Each driver may also be modeled with different *nervousness* factors by changing the acceleration or the safety distance it will employ and consequently the dangerousness of its driving.

However, VanetMobiSim is not able to model *malicious* drivers potentially creating accidents (such as crossing at a red light), considering the fact that it is currently unable to model accidents. Indeed, the first step before including such drivers in VanetMobiSim would be to model the impact of its actions, the lack of which would undermine the impact of such malicious driving. Considering the modularity of the structure of VanetMobiSim, such new module could easily be added once a specific traffic model is defined. We leave this approach as future work.

5. VanetMobiSim Validation

In this section, we validate the realism of the vehicular movement traces produced by various mobility models included in VanetMobiSim[3]. Our objective is to verify that the overall mobility description provided by VanetMobiSim, notably thanks to its original features, is able to model vehicular traffic with a sufficient level of realism. We first compare the traces obtained with different mobility models implemented in VanetMobiSim and show that its original model involving car interactions and intersection management are more realistic. Second, we formally validate the traces of VanetMobiSim against a benchmark traffic simulator, therefore guaranteeing the level of realism of VanetMobiSim's traces.

First, different micro-mobility models are tested on a user-defined graph representing a square city section of 1,500 m side. The urban topology employed in those tests is shown in Figure 11, where, unless specified differently, all roads have a single lane and a speed limit of 15 m/s (54 km/h), except for the roads represented with thicker lines that allow a maximum speed of 20 m/s (72 km/h). Vehicles travel between entry/exit points at borders, identified with circles and squares, and are crossing the city section according to the fastest path to their destination. The trips generation scheme is activity based (see Section 4.4.1) and the relative transition probability matrix describes a simple activity chain depicted in Figure 12. The different states denote the class of the selected destination: a round for the entry/exit points of high-speed roads, a square for the entry/exit points of normal-speed roads as also shown in Figure 12. The chain is trivially

---

[3] Throughout this section, the IDM-IM and IDM-LC are VanetMobiSim's original models, while the others are inherited from CanuMobiSim.

Table 2. Parameter values for RWP, CSM, and FTM microscopic mobility models.

Model	RWP		CSM		FTM		
Parameter	speed	pause	speed	pause	$s_{\min}$	$s_{\max}$	κ_{jam}
Value	$unif 10, 20$ m/s	$unif 0, 60$ s	$unif 10, 20$ m/s	$unif 0, 45$ s	3 m/s	20 m/s	0.125 car/m

Table 3. Parameter values for IDM, IDM-IM and IDM-LC microscopic mobility models

Model	IDM					IDM-IM	IDM-LC		
Parameter	v_0	s_0	T	a	b	κ	a_{bias}	p	a_{thr}
Value	$unif 10, 20$ m/s	1 m	0.5 s	0.6 m/s^2	0.9 m/s^2	5	0.2 m/s^2	0.5	0.2 m/s^2

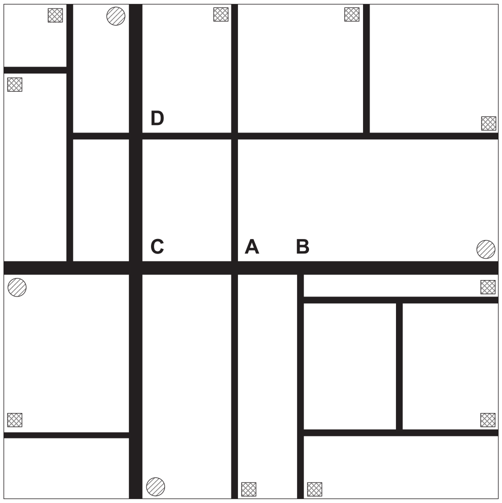


Figure 11. City section topology.

ergodic, with steady state $(p/(p+q), q/(p+q))$. In our simulation, the probabilities are set so that $p = q = 1/2$, resulting in a stationary distribution $\left(\frac{1}{2}, \frac{1}{2}\right)$. This, along with the proportion between the number of entry/exit points of the two classes, determines a popularity of high-speed roads entry/exit points which is more than double with respect to that of normal-speed entry/exit points. This mimics the tendency of traffic flows to concentrate on the main, high-speed roads.

The number of cars traveling at the same time within the city section ranges from 100 to 500, reproducing light (10 vehicles/km) to heavy (50 vehicles/km) traffic conditions. For each test, a single simulation was run, with statistics recorded for 3,600 s after a transient phase of 900 s[4]. When computing 95% confidence intervals for mean values averaged in time and on the whole road topology, the error margin was found to be within 0.5% from the mean.

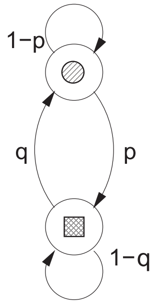


Figure 12. Activity chain.

The parameters of the mobility models used in these experiments are listed in Tables 2 and 3. Owing to space limitations, we cannot provide an in-depth discussion of the models calibration, and invite the interested reader to refer to the models' references for details. We just stress that the selected parameters fit real-world values and that we calibrated them according to the scenario.

5.1 Comparison of the Vehicular Mobility Models

In this section, we validate VanetMobiSim by showing how it implements mobility models which are able to produce traces which are more realistic than those obtained with popular vehicular mobility models. Among those, we also consider the Random Waypoint Model (RWP), in order to illustrate the 'worst case' (in terms of realism) scenario for vehicular mobility. Indeed, owing to its nature, this model is not bound by motion constraints or driven by a traffic generator.

In Figure 13, the trend of the average speed versus the number of vehicles is shown. RWP and CSM ignore car-to-car interactions and are not affected by the number of vehicles present on the topology, which leads to an unrealistically constant mean speed. The mean velocity recorded with CSM is slightly lower than that measured with RWP, even if the mean pause time is shorter in CSM than in RWP. The reason is that CSM limits the nodes movement to the road topology, with pauses at each intersection en-

---

[4] VanetMobiSim offers the possibility to exclude mobility traces obtained before a configurable simulation time, independently of the trace file format.

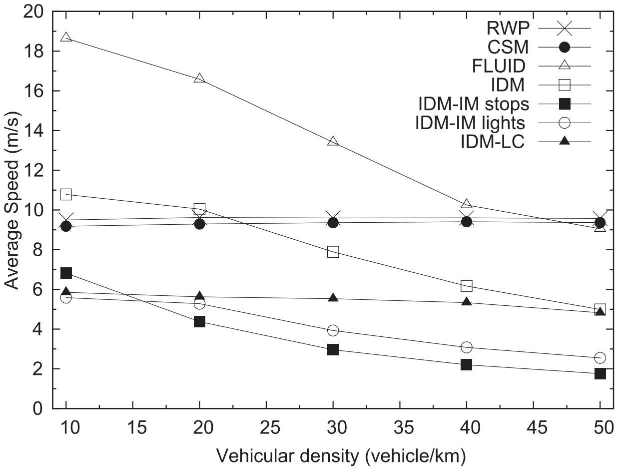


Figure 13. Average speed versus vehicular density.

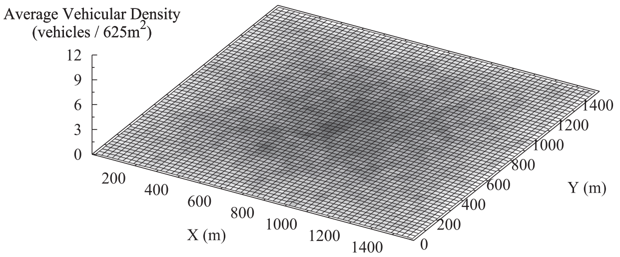


Figure 14. Vehicular density: RWP.

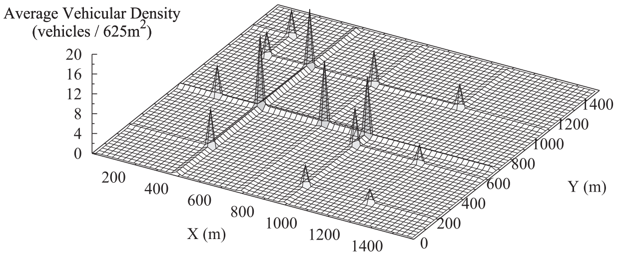


Figure 15. Vehicular density: CSM.

countered on the path. Thus, the average distance between subsequent pauses is reduced in CSM at the consequence of a lower average speed.

The low level of realism of these models is further evidenced in Figures 14 and 15, depicting the time-averaged vehicular density distributions over the road topology obtained with RWP and CSM, respectively. These distribution plots, as well as the equivalent plots for the other mobility models in the remainder of this section, refer to the 30 vehicles/km case.

As expected, RWP spreads nodes all over the square area, yet with a higher density of nodes in the center of the map, as part of RWP normal behavior [29].

On the other hand, cars driven by CSM follow the road topology, and we can observe a non-zero density only where roads are present. Also, in Figure 15 the effect of the activity-based mobility can be observed: the two faster and more frequented roads experience a higher vehicular density with respect to the other streets in the topology. The same can be observed also in the vehicular density plots obtained with the other micro-mobility models. However, CSM produces what we call an *on–off* behavior, with a constant vehicular density on roads and sudden high peaks (note the different node density scale with respect to the equivalent plots of the other micro-mobility models) at intersections, where vehicles overlap and stop for a random amount of time. The absence of car-to-car interaction leads thus to an unrealistic complete absence of queuing or acceleration/deceleration phenomena in the proximity of intersections.

Looking back at Figure 13, modeling the vehicular mobility with FTM produces a very high average speed, which is mostly due to the fact that vehicles never stop with this model, as the zero speed condition would cause a deadlock in the model described in Section 4.4. A smaller value of the κ_{jam} parameter would probably have reduced this effect, producing a lower and more realistic figure of the average velocity. However, the settings we chose force vehicles to move at a minimum speed of 10 km/h when they are at a distance of 3 m or less from each other, which represents a suitable real-world condition. As expected, FTM reproduces the average speed reduction caused by the vehicular density growth, since the increase of the number of cars traveling concurrently on the same road reduces the fluid speed. However, the vehicular density distribution depicted in Figure 16 demonstrates the non-sufficient realism of this model. In the considered scenario, a high density is experienced by the central segment marked as AB in Figure 11, which is shared by many of the possible paths drivers can choose from. The high quantity of cars driving through determines a reduction of the speed according to the model and creates an even higher vehicular density. Although that is consistent with what would happen in a real-world situation, FTM reasons on a per-edge basis and produces a constant car density over each street, which results in the absence of traffic correlation over connected roads. In our case, it can be noticed that the high car density in AB suddenly disappears in roads after intersections A and B (see Figure 11 for the mapping of letters to intersections). Moreover, as FTM ignores intersections, the average number of vehicles at crossroads does not differ from that of vehicles on roads nearby, which again, is far from reality.

As far as IDM is concerned, the average speed curve in Figure 13 shows lower values when compared with that obtained with FTM and, quite surprisingly, appears to be affected by the number of cars present on the topology.

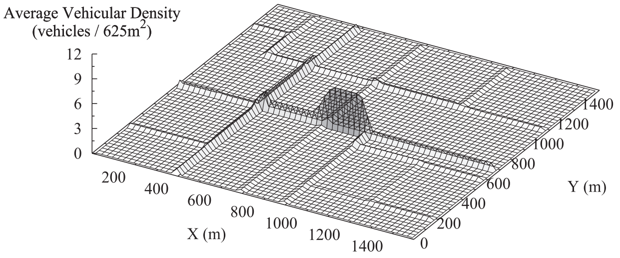


Figure 16. Vehicular density: FTM.

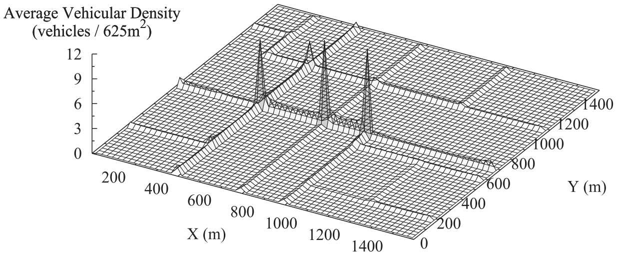


Figure 17. Vehicular density: IDM.

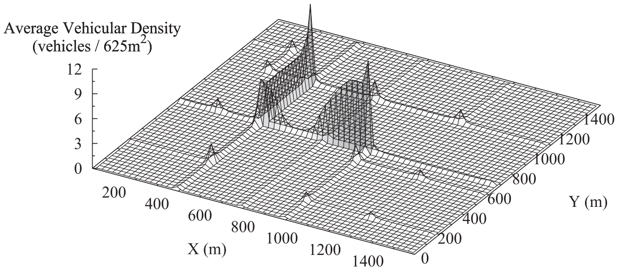


Figure 18. Vehicular density: IDM-IM, stop signs.

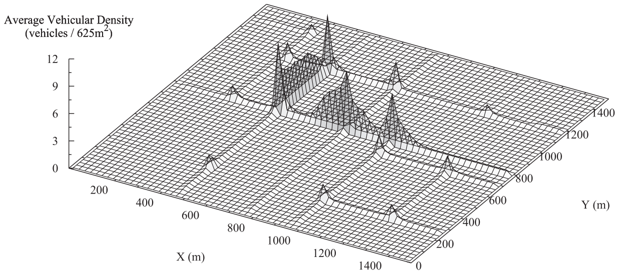


Figure 19. Vehicular density: CORSIM, stop signs.

The speed reduction with respect to FTM is imputable to a more realistic car-to-car interaction, which leads to queuing of fast vehicles behind slow cars. The dependence from vehicular density has instead a two-fold nature: first, the higher density increases the probability of encountering slow vehicles, which generate queues and force a reduction on other drivers' speed. Second, there exists a side effect of the CanuMobiSim implementation, that occurs when vehicles coming from different directions and overlapping at intersections suddenly notice that the safety distance condition is violated. According to the current implementation, they stop and wait for a distance s_0 to be restored before leaving the junction. Such a situation causes the average speed to decrease, and occurs more and more frequently as the vehicular density grows. In Figure 17, the vehicular density proves that the realism of an accurate car-to-car interaction model in urban scenarios is low, if intersection management is not taken into account. Spikes at highly frequented intersections A, B and C are to impute to the implementation issue explained above. Accordingly, we can state that IDM does not perform more realistically than FTM in an urban context.

Two different tests were run for IDM-IM, the first original model provided by VanetMobiSim. The first contains intersections regulated by stop signs, while the second includes traffic lights at road junctions. As observed in Figure 13, the model produces a very low average speed, since cars spend most of their time queued at intersections. The problem is exacerbated as the density of vehicles increases and causes longer queues. This can also be noticed by looking at the vehicular density in Figure 18, where high vehicular densities, accounting for long queues, are recorded in the neighborhoods of the main intersections A, B, C and D. The higher concentration of vehicles around these intersections also has the side-effect of reducing the number of vehicles on the other roads of the topology, which, as a matter of fact, record lower vehicular densities. A realistic effect of smooth vehicular increasing density towards the congested crossroads is obtained with this model. It can be noticed that such an effect is not limited to single segments as it happened with FTM, but also has an impact on adjacent roads.

When traffic lights with a period of 90 s are used to regulate traffic at intersections, vehicular mobility is improved with respect to the stop sign case, especially in dense scenarios (see Figure 13). This could be expected, as traffic lights replace the slow 'taking-turns' crossroads management induced by stop signs with a faster 'burst' mechanism, in which groups of cars are allowed to cross the junction one after the other, and are able to save on acceleration delay. However, for the same reason observed in the stop sign case, the mean speed is still reduced when more cars are introduced into the road topology. An interesting effect can be observed when the vehicular density is low: the stop sign case outperforms the traffic light case. This occurs because,when the number of cars is small, the probability that a crossroad is free is high. Thus, the passage is often immediately granted with a stop handling

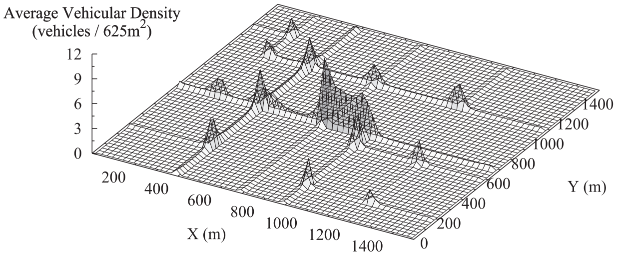


Figure 20. Vehicular density: IDM-IM, traffic lights.

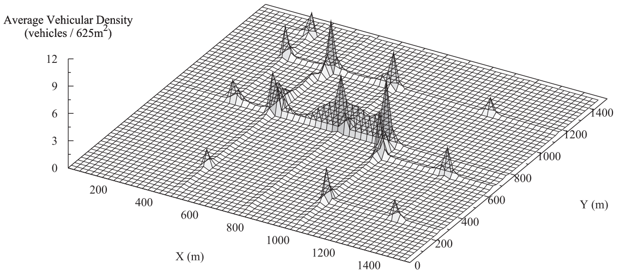


Figure 21. Vehicular density: CORSIM, traffic lights.

of intersections, but at the cost of slowing down and accelerating again. On the other hand, when a traffic light management is considered, vehicles still have to stop in the presence of red traffic lights and wait for the light to turn green, even if there are no other cars waiting to cross the intersection. This is a pattern that can be naturally observed in real urban scenarios. The vehicular density presented in Figure 20 is reduced, as queuing at highly visited intersections is still present, but noticeably reduced with respect to the previous IDM-IM scenario with stop signs. Thanks to the improved distribution of traffic over the whole topology, the queuing phenomenon can now be observed at minor intersections, where vehicles have to wait for green traffic lights.

In Figures 22 and 23, the mean vehicular speed recorded on the road topology is presented, when IDM and IDM-IM in conjunction with stop signs at intersections are respectively used. The darker the color, the lower the average speed of vehicles traveling on it, down to zero speed for black areas (such as those outside the road topology). By looking at these figures it is clear that the presence of a model accounting for the car-to-infrastructure interaction can noticeably affect the outcome of the simulation. Indeed, if such an aspect is ignored, even in presence of accurate car-to-car analysis, as is the case in Figure 22, the speed distribution is almost constant all over the road topology, with the exception of main intersection points where the speed is slightly lower due to the IDM safety distance management issue discussed before. A constant speed behavior can be observed at minor intersections, where cars never stop. On the other hand, when the vehicle behavior at road junctions is modeled correctly, as in Figure 23, the speed figure over the road topology appears far more realistic, with decelerations and queuing when approaching the junctions, and accelerations when leaving them. These figures also depict the higher average speed of vehicles moving on the high-speed central roads, represented by brighter surfaces.

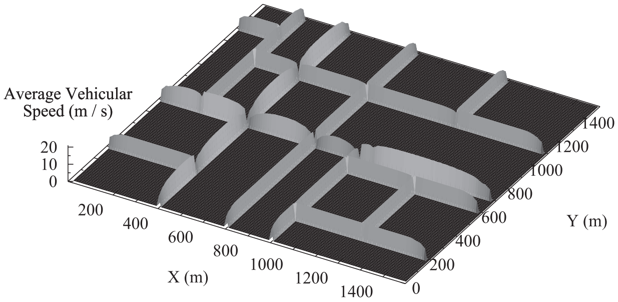


Figure 22. Speed distribution: IDM.

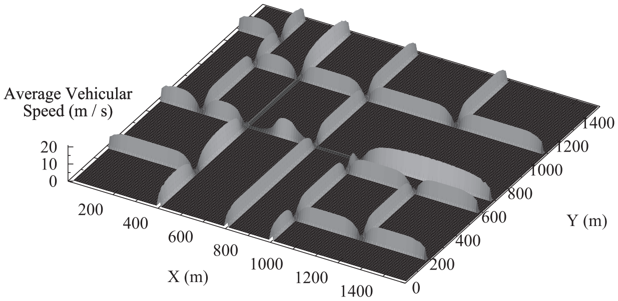


Figure 23. Speed distribution: IDM-IM, stop signs.

Finally, we report the results obtained when IDM-LC, VanetMobiSim's second original mobility model, is employed as the micro-mobility model. We considered traffic lights at intersections and two per-direction lanes on each road. From Figure 13, modeling vehicular micro-mobility with IDM-LC seems to avoid most of the speed decay effects discussed previously. This is an interesting result, motivated by the fact that (i) vehicles actually employ overtakings to avoid slow cars and congested lanes, thus increasing the average velocity, and (ii) the presence of multiple lanes helps vehicular mobility in the presence of densely populated intersections, as multiple cars can pass through the intersection at the same time and reduce the bottleneck effect of road junctions. In other words, the availability of two parallel unidirectional lanes on each road does not only physically double the capacity of the urban infrastructure, leading to a halved perceived vehicular density, but also brings important correlated effects. This effect may also be naturally observed in real urban

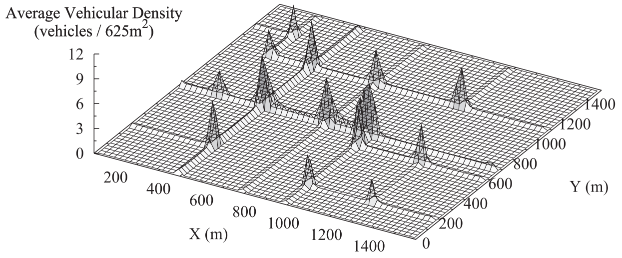


Figure 24. Vehicular density: IDM-LC.

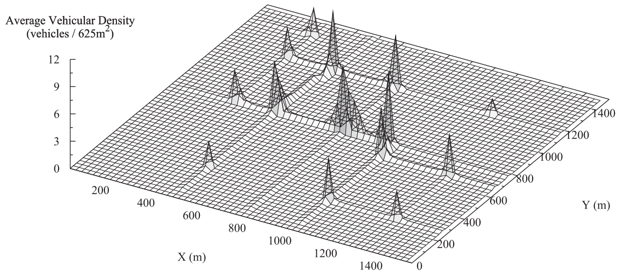


Figure 25. Vehicular density: CORSIM, lane changes.

scenarios. For the reasons explained before, the maximum simulated density of 50 vehicles/km would appear, in our case, as a density of less than 25 vehicles/km. This traffic condition therefore does not seem to generate severe traffic congestion. The vehicular density measured with IDM-LC is depicted in Figure 24 and shows that queuing phenomena at intersections are almost equally distributed over the whole topology. Minor intersections experience a higher density with respect to the IDM-IM case as, in the absence of critical congestion situations at main junctions, vehicles are more uniformly spread and their presence at smaller crossroads is more noteworthy.

In a different test, we exploited the vehicular mobility description provided by VanetMobiSim to recreate a validating scenario in vehicular traffic analysis. In Figure 26, the shock waves produced on vehicular density by a periodic perturbation are shown. This result has been obtained with IDM-LC on a 1 km long, unidirectional, double-lane, and straight road. Cars move towards positive abscissae and a traffic light, located halfway and with a period of 360 s, is used as the perturbation source. We can notice that the red traffic light inhibits the movement of vehicles, causing them to stop at 500 m. As shown by the increasing vehicular density, a queue is formed as more vehicles approach the traffic light. However, when the traffic light turns green, queued vehicles start flowing towards and through the second half of the road. It is possible to see that the high-density shockwave propagates in the opposite direction with respect to movement of cars as time goes on. The speed dynamics recorded during the same experiment are depicted in Figure 28, where we can better observe the queuing perturbation, which are represented by the dark and zero-speed areas, propagating against the traffic flow direction in time. Shock waves are a common phenomena of real-world traffic. When long queues form in the proximity of perturbation sources (crowded intersections, toll stations, in-flow ramps, etc.) the finite reaction time of drivers determines a delay in the propagation of movement. Thus, vehicles queued far from the perturbation origin experience changes in velocity or local traffic density only a long time after the original mobility change occurs at the perturbation.

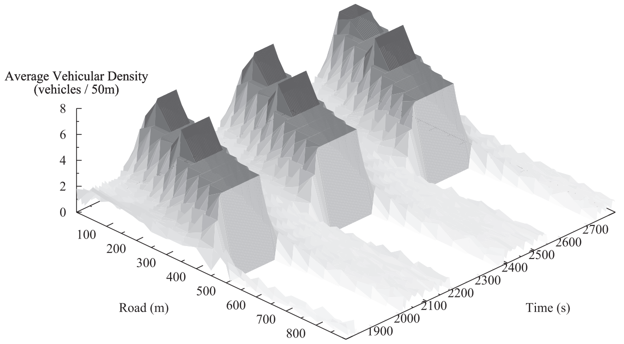


Figure 26. Vehicular density shock waves: IDM-LC.

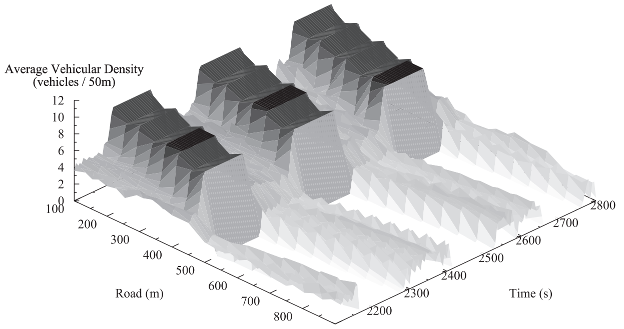


Figure 27. Vehicular density shock waves: CORSIM.

In this section, we illustrated how the traces produced by VanetMobiSim's original microscopic models, improved by realistic mobility constraints and traffic generators, were able to outclass mobility models previously available. Unlike those models, the IDM-IM and IDM-LC are able to model realistic vehicular motion patterns.

5.2 Validation Against a Benchmark: TSIS-CORSIM

Now that we have proved that VanetMobiSim's models are able to reproduce mobility patterns which are *intuitively* more realistic than those obtained with other common-use models, one may now raise the question of *how* realistic

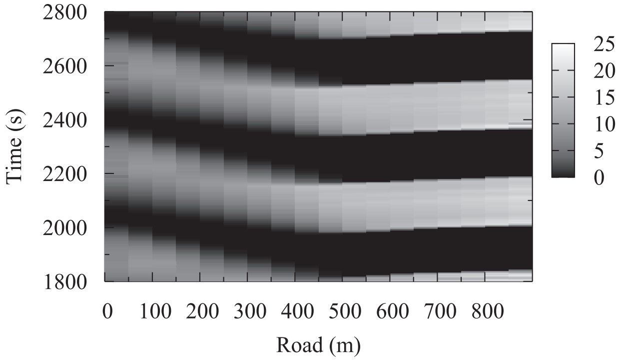


Figure 28. Vehicular speed shock waves: IDM-LC.

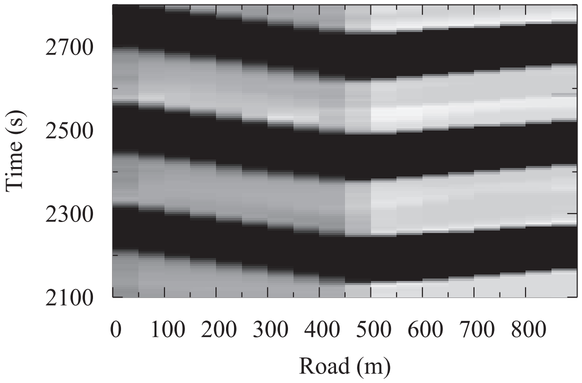


Figure 29. Vehicular speed shock waves: CORSIM.

our traces are. One solution in order to verify the realism of a mobility model would be to compare its synthetic traces with real mobility traces. However, to the best of our knowledge, no urban traffic traces are publicly available, apart from a few referring to public transportation systems [24, 25], which are not representative of typical vehicular mobility. Thus, the only solution is to compare our synthetic traces with those obtained from a traffic simulator which has already been calibrated and validated. To this extent, in this section we test VanetMobiSim against TSIS-CORSIM, a benchmark traffic simulator widely employed in the traffic engineering community.

TSIS-CORSIM [14] is a comprehensive traffic simulator applicable to surface streets, freeways, and integrated networks with a complete selection of control devices (i.e. stop/yield sign, traffic signals, and ramp metering). It simulates traffic and traffic control systems using commonly accepted vehicle and driver behavior models. CORSIM has been validated by showing its ability to generate mobility traces identical to real traces gathered in predefined testing areas. CORSIM has been employed by thousands of practitioners and researchers worldwide over the past 30 years and embodies a wealth of experience and maturity. Funded by the US Federal Highway Administration (FHWA) throughout the last three decades, TSIS-CORSIM has evolved into a benchmark tool for transportation planning professionals. We validated VanetMobiSim against CORSIM version 5.1.

Before moving further, we would like to answer to a question which might appear obvious: if CORSIM is able to provide realistic and validated vehicular motion patterns, why do we need VanetMobiSim?

In order to answer this question, we have to understand what CORSIM is and what is its intended target. CORSIM has been created for transportation, traffic, and civil engineers. As networking was completely out of the scope of those targets, no interface has been created for interacting with external simulators. CORSIM does not output any other data than statistics. Moreover, it communicates with the visualization tool TRAFVU using a set of byte code files that cannot be easily interpreted. Last but not least, copyrights forbid anyone from changing this significant limitation in order to add an interface with external simulators. The only solution to produce CORSIM output traces that could be interpreted by network simulators is to create a specific parser to extract vehicular mobility information from the byte code files. More recent tools such as AIMSUN [16] or VISSIM [15] provide raw output files that are easier to parse and therefore partially solve this issue. A second significant limitation of CORSIM, that is unfortunately also similar to VISSIM and AIMSUN, is the complexity of its calibration. The level of realism involved in traffic and transportation planning is so high that a potential CORSIM user needs to tweak approximately 30 parameters, each influencing vehicular traffic, in order to calibrate the simulation. This process requires knowledge of traffic theory basics, is very time consuming and, for the greater part, appears useless to networking researchers, as network simulation does not require such a level of detail of vehicle mobility. VanetMobiSim does not intend to reproduce vehicular traffic with the same level of precision, and thus only needs approximately 5 to 10 parameters as it is rather tailored to the networking community. The last limitation of CORSIM is its commercial nature. Indeed, unlike VanetMobiSim, CORSIM is not free, and copyrights block any contribution from the vehicular communication community to improve it. For all of these reasons, we believe that VanetMobiSim brings a sufficient level of realism for vehicular communications, is easier to configure than CORSIM, does provide a straightforward interface with various network simulators, and is finally open source and free.

We configured CORSIM according to the urban topology and activity chain in Figures 11 and 12. As CORSIM has been designed to model urban traffic with a high level of precision, it also contains a large set of configuration parameters. For parameters common to VanetMobiSim and CORSIM, we used the same values (see Table 3), while for parameters unique to CORSIM, we kept the software's default values.

For sake of clarity, vehicular densities obtained with CORSIM are matched with those obtained with VanetMobiSim in Figures 19, 21 and 25. As it is easily noticed, the evolution of traffic in the three cases of stop signs, traffic lights, and multiple lanes is equivalent in VanetMobiSim and CORSIM, and the comments on the reasons behind these densities provided in Section 5.1 while referring to the VanetMobiSim traces also holds for the CORSIM output.

We stress that as CORSIM receives the vehicular inflow, and not the number of nodes, as an input, the exact number of cars simulated by CORSIM cannot be easily controlled, and there is no guarantee that the number of cars is identical in VanetMobiSim and CORSIM simulations: this leads to slight differences in local vehicular densities. However, we are more interested in the geographical distribution of the cars than in local intensity, and such a distribution appears consistent between the simulators.

Comparable results are returned by the two tools also in presence of a shockwave effect created by a periodic perturbation, as shown in Figures 27 and 29. The process through which these plots are obtained with CORSIM is identical to that employed in the VanetMobiSim case and described earlier in Section 5.1.

To conclude this section, we would like to emphasize that although neither CORSIM or VanetMobiSim use similar mobility models, nor are controlled by the same set of configuration parameters, the traffic distribution and the shockwaves phenomenon reproduction obtained with the two tools are similar and conform to real-life situation. This is a proof of the good level of realism reached by the mobility traces generated by VanetMobiSim, which can thus be confidently employed to reproduce everyday vehicular motion patterns.

6. Related Work

The world of vehicular or traffic simulations is mostly bipolar. On the one side are the professional and commercial license-based tools while on the other side are the free and open-source solutions. It is not trivial to judge one or the other as both sides have their advantages and drawbacks. For example, the professional tools are usually very realistic, to a level that is mostly not required for network simulations. However, as the sources are not available, it is not possible to develop new experimental modules not included in the original version. Moreover, expensive commercial licenses are usually a dissuading factor in research. On the other side, the open-source tools are freely available but significantly suffer from a limited level of realism compared with professional tools, although the recent years have witnessed a significant improvement in realism. In this section, we review related research works focusing on modeling vehicular mobility for VANETs simulation.

On the side of professional tools, many realistic traffic simulation tools, such as PARAMICS [13], CORSIM [14], AIMSUN [16], VISSIM [15] or TRANSIMS [17], have been developed to analyze vehicular mobility at both microscopic and macroscopic levels with a very high degree of detail. However, the aforementioned tools are distributed mostly under commercial licenses which is a major impediment to adoption by the academic research community. Moreover, the presence of copyrights or simply the unavailability of source files impedes the modification/extension of these tools when particular conditions, not planned by the original software, have to be implemented and simulated. These tools also suffer from similar issues to those described for CORSIM in the previous section. With the exception of a few teams that developed parsers [18, 19] or federated a realistic traffic simulation tool with a network simulator [20, 63], these tools have been originally designed for traffic analysis and not to generate movement traces usable by networking simulators. Some tools, however, such as VISSIM and AIMSUN, are able to output trace files that can later be converted to input files for network simulators. For a review and comparison of commercial traffic simulation tools, the interested reader can refer to [21].

On the side of free tools, purely random models, such as the RWP model, the Random Walk model, the Reference Point Group (or Platoon) model, the Node Following model or the Gauss–Markov model, just to cite the most well known, are often used when simulating MANETs mobility. However, when VANETs are considered, employing these models usually produces unreliable results, as they do not even pose fixed road constraints on the motion of the nodes. The simple Freeway model and Manhattan model [22] represent an initial step in mimicking vehicular movements, while several pioneering works in this field involve the generation of mobility patterns based on real road maps [23] or monitoring of real vehicular movements in cities [24]. In most of these models, only the macro-mobility of nodes is considered, although car-to-car interaction is a fundamental factor to take into account when dealing with vehicular mobility [26]. Further details on many of these models can be found in [27–29].

Several open-source tools for the generation of vehicular mobility patterns became available in recent years. In the rest of this section, we briefly review the most well known, but an extended survey and comparison of such tools can be found in [30].

Most of these open-source tools are capable of producing traces for network simulators such as *ns-2* [31], *QualNet* [33] or *OPNET* [34]. The IMPORTANT tool [22], and the BonnMotion tool [35] implement several random mobility models, including the Manhattan model. They both focus on macro-mobility, as IMPORTANT only includes a basic car-to-car inter-distance control schema called *Car Following Model*, whereas BonnMotion does not consider any micro-mobility. Consequently, the output traces are definitely too simple to represent realistic motion. Within the MONARCH project, a tool to extract road topologies from real road maps obtained from the TIGER database [36] was realized [23], but, again, the lack of micro-mobility support makes it impossible to reproduce real-world vehicular mobility.

The Mobility Model Generator for Vehicular Networks (MOVE) [44] adds TIGER map parsing and trace generation capabilities to SUMO [45], an open-source vehicular macro- and micro-mobility simulator. The MOVE project started and progressed alongside our own, but seems today abandoned. A new project, eWorld [46], is a traffic simulator front-end to configure SUMO and is able to import, edit, and enrich street maps from the OpenStreetMap [47] database, visualize the vehicular mobility traces generated by SUMO and provide visual statistics. Considered together, SUMO and eWorld reach the same user-friendly configuration objectives as the latest version of VanetMobiSim.

The Street Random Waypoint (STRAW) tool [48] is a mobility simulator based on the freely available Scalable Wireless Ad Hoc Network Simulator (SWANS). STRAW is able to parse TIGER files and also implements complex intersection management using traffic lights and traffic signs, but its dependence on SWANS prevents research communities adopting different network simulators from using it. A similar solution has been adopted by the team of VGrid [49]. They developed a simulation tool federating the network simulator SWANS and a synthetic traffic model. The complex vehicular flows are based on the Nagel and Schreckenberg model, extended to include lane changing in highway scenarios. The network simulator and the traffic simulator interact with each other by means of specific input and output messages The GrooveSim tool [50] is a mobility and communication simulator, which again uses files from the TIGER database to simulate realistic road topologies. However, GrooveSim does not model vehicle micro-mobility, or produces traces that are usable by network simulators. The City Model [51], although considering car-to-car interactions, falls short of producing a realistic mobility, mainly due to a poor grid-based macro-mobility description. The SSM/TSM model [52] includes a *Stop Sign Model* and a *Traffic Sign Model*, but it only considers a simplified intersection behavior, whereas no car-to-car mobility seems to be taken into account when vehicles are traveling on roads. MobiREAL [53], although appearing mainly focused on modeling pedestrian mobility, introduces a novel approach to cognitive modeling which could be promising in the direction of future extension to vehicular mobility.

Recent works have adopted an embedded approach, joining a scalable vehicular mobility description and network stack modeling in a single simulation tool. MoVes [54] is a complex mobility generator, built on top of ARTiS [55], a scalable distributed simulation middleware. MoVes features cars following models, drivers behavior characterization, intersection management and includes a parser module to include GPS maps using the GPS TrackMaker program [56]. At the current stage of development, MoVes does not support lane changing and realistic path generation. Also [57] presents an ongoing project on an integrated network and mobility simulator, featuring a TIGER maps parser and a VISSIM-like micro-mobility model [15]. Another tool possibly falling into the embedded category is NCTUns [58], which is not focused on vehicular mobility, and provides a full range of network stack simulation/emulation tools. Vyyuru et al. instead proposed AutoMesh [59], a realistic simulation framework for VANETs. It is composed of a set of modules controlling all parts of a realistic simulation. It includes a *driving simulator module*, a *radio propagation module*, and *a network simulator module*, all interlinked with feedback in order that any alterations made in one module have an influence on the other modules. At the stage of the development of AutoMesh, the driving simulator module only includes random macro-movement and the IDM model for micro-movements. It is therefore unable to reproduce the non-uniform distribution of positions and speed usually experienced in urban areas. However, the radio propagation module is very detailed, using 3D maps and digital elevation models in order to obtain a realistic radio propagation model in urban area. All of these combined approaches have the big advantage of allowing a direct interaction between the communication network system and the vehicular traffic dynamics, so that the first can have an influence on the second. However, two major flaws are induced by this *modus operandi*. First, the level of detail of both modules is necessarily lower than that provided by ad-hoc simulation tools. This is especially true for network simulation modules, which are nowadays required to model a variety of protocols, mechanisms, and physical phenomena with a high degree of confidence, and even successful, dedicated, open-source projects, which enjoy the contribution and support of thousands of users, struggle to reach such a goal. Thus, building and validating new simulators from scratches may result in reduced modeling capabilities and/or low realism. This is also at the basis of the second disadvantage of a joint simulation of mobility and network: in order to benefit from the mobility description, the embedded network simulator must be used. This prevents the largest part of networking research, used in common and reliable ad-hoc network simulators, from adopting these tools, therefore limiting their diffusion.

UDel Models [61] is a suite of tools for the generation of urban mobility and the computation of radio propagation, intended for large-scale urban mesh networks. The vehicular and pedestrian mobility description is significantly different from that presented in all of the aforementioned work, as it is based on statistical data from real-world measurements and surveys, mainly obtained from databases of the US Department of Labor. The urban propagation model includes an accurate map builder capable of parsing GIS data and computing from that realistic radio signal propagation, and represents a valuable addition to the mobility description. However, as far as vehicular mobility is concerned, it is not clear which is the level of detail in the macro- and micro-mobility models.

The SHIFT traffic simulator [60] has been developed within the PATH project, and is now a well-established micro-mobility simulator that generates trajectories of vehicles on highway scenarios according to validated mobility models.

Recently, the growing interest in a stronger interaction between vehicular mobility and networking simulation, allowing communication between cars to affect and skew the traffic distribution, led to several efforts aimed at merging existing ad-hoc simulators. These attempts include TraNS [62], merging SUMO and ns-2, and MSIE [63], employing VISSIM, ns-2, and Matlab. The first project is under development, while the second encountered a very limited diffusion as it uses the commercial VISSIM for the mobility modeling part. VanetMobiSim currently does not support this approach as it solely provides vehicular mobility traces. However, the interlinking of VanetMobiSim with a network simulator is currently being investigated as we believe that this feature will be crucial in the future use of vehicular mobility simulations. It is however independent of the vehicular mobility modeling itself and thanks to VanetMobiSim modularity, such enhancement is totally feasible.

As a final remark, we would like to point out that, in most of the works mentioned in this section, small or no attention is paid to the validation of the macro- and micro-mobility models used, and evaluations on the realism of the produced traces are often absent. The spirit behind VanetMobiSim is to develop a freely available open-source tool, such that the community can add new experimental features, and which level of realism could compete with professional tools.

7. Conclusions

In this paper we have presented VanetMobiSim, an open-source extension to the CanuMobiSim user mobility framework, capable of producing realistic vehicular mobility traces for several network simulators. We first illustrated the need for realism in vehicular mobility modeling, as most of the simplistic models available nowadays for network simulation do not pass standard tests from traffic theory. We also presented a concept map and derived guidelines for the generation of realistic vehicular mobility models. We then reviewed the software and data model, the motion constraints and traffic generator functional blocks of VanetMobiSim, and provided details of the significant additions brought to CanuMobiSim in order to develop realistic vehicular-specific motion patterns. As for the following steps, we compared the traces obtained with VanetMobiSim's original models, the IDM-IM and IDM-LC, against those produced by common-use mobility models, considering metrics such as the vehicular density and speed distribution, as well as applying standard traffic theory tests. Finally, we proved that the mobility traces obtained with VanetMobiSim are consistent with those generated by a benchmark, widely used, validated traffic simulator called TSIS-CORSIM.

By taking a comprehensive look at the results obtained, it appears clear that the level of detail of various mobility models commonly employed nowadays in networking research is not sufficient to reproduce realistic vehicular traffic traces. The increased degree of detail introduced in VanetMobiSim by the new IDM-IM and IDM-LC models in the traffic generator block, and the possibility of their interaction with the new mobility constraints functional block, appears necessary to reproduce real-world phenomena. Thus, we believe that the public-availability of VanetMobiSim sources to the vehicular networking research community is a valuable contribution to the progress of studies in this area.

8. Acknowledgments

We would like to thank Professor Mario Gerla and Dr Giovanni Pau from the Network Research Lab at the University of California in Los Angeles (UCLA) for having provided us the access to TSIS-CORSIM during our stay.

Dr Jérôme Härri acknowledges the support of the Ministry of Science, Research and the Arts of Baden-Württemberg (Az: Zu 33-827.377/19,20), the Klaus Tschira Stiftung, the INIT GmbH and the PTV AG for the research group on Traffic Telematics.

9. References

[1] Fiore, M., J. Härri, F. Filali and C. Bonnet. 2007. Vehicular mobility simulation for VANETs. In *Proceedings of the 40th Annual Simulation Symposium (ANSS '07)*.

[2] US Department of Transportation (DoT) IntelliDrive, http://www.intellidriveusa.org/.

[3] Car-2-Car Communication Consortium (C2C-CC), http://www.car-to-car.org/.

[4] IEEE Wireless Access for the Vehicular Environment (WAVE), http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm.

[5] ETSI Technical Committee on Intelligent Transportation Systems (ETSI TC ITS), http://www.etsi.org.

[6] ISO Technical Committee 204 Working Group 16, Continuous Communication for Vehicles (CALM), http://www.isotc204wg16.org.

[7] VanetMobiSim Project, http://vanet.eurecom.fr.

[8] CANU Project, http://canu.informatik.uni-stuttgart.de.

[9] Haerri, J., F. Filali and C. Bonnet. 2005. A framework for mobility models generation and its application to inter-vehicular networks. In *3rd IEEE International Workshop on Mobility Management and Wireless Access (MobiWac'05)*.

[10] Lighthill, M.J., and G.B. Whitham. 1955. On kinematic waves II: a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society A*, 229: 317–345.

[11] Triterer, J. 1975. *Investigation of Traffic Dynamics by Aerial Photogrammetry Techniques*. Technical Report 278, Transportation Research Center, Ohio State University.

[12] The SUN Java Platform, ns-2, http://java.sun.com/javase/6/.

[13] Paramics: Microscopic Traffic Simulation, http://www.paramics-online.com.

[14] CORSIM: Microscopic Traffic Simulation Model, ttp://mctrans.ce.ufl.edu/featured/tsis.

[15] ptv simulation – VISSIM, http://www.english.ptv.de/cgi-bin/traffic/traf_vissim.pl.

[16] TSS AIMSUN, http://http://www.aimsun.com.

[17] TRANSIMS, transims.tsasa.lanl.gov.

[18] Yin, J., T. El Batt, G. Yeung, B. Ryu and S. Habermas. 2004. Performance evaluation of safety applications over DSRC vehicular ad hoc networks. In *ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2004)*, Philadelphia, PA, October 2004.

[19] Wu, H., R. Fujimoto, R. Guensler and M. Hunter. 2004. MDDV: Mobility-Centric Data Dissemination Algorithm for Vehicular Networks. In *ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2004)*, Philadelphia, PA, October 2004.

[20] The Federated Simulations Development Kit (FDK), http://www-static.cc.gatech.edu/computing/pads/fdk.html.

[21] Boxill, S. and L. Yu. 2000. *An Evaluation of Traffic Simulation Models for Supporting ITS Development*. Technical Report 167602-1, Texas Southern University, October 2000.

[22] Bai, F., N. Sadagopan and A. Helmy. 2003. The IMPORTANT framework for analyzing the impact of mobility on performance of routing for ad hoc networks. *Ad Hoc Networks Journal*, 1(4): 383–403.

[23] Saha, A.K., and D. Johnson. 2004. Modeling mobility for vehicular ad hoc networks. In *Poster Session at the 1st ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004)*, Philadelphia, PA, October 2004.

[24] Jetcheva, J., Y.-C. Hu, S. PalChaudhuri, A.K. Saha and D.B. Johnson. 2003. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *5th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2003)*, Monterey, CA, October 2003.

[25] The CRAWDAD database, Traces of the fleet of city buses in Seattle, http://crawdad.cs.dartmouth.edu/meta.php?name=rice/ad_hoc_city.

[26] D. Helbing. 2001. Traffic and related self-driven many-particles systems. *Reviews of Modern Physics*, 73: 1067–1141.

[27] Camp, T., J. Boleng and V. Davies. 2002. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5): 483–502.

[28] Bai, F., and A. Helmy. 2004. A survey of mobility modeling and analysis in wireless ad hoc networks. In *Wireless Ad Hoc and Sensor Networks*, Kluwer, Dordrecht.

[29] Le Boudec, J.-Y., and M. Vojnovic. 2005. Perfect simulation and stationarity of a class of mobility models. In *IEEE INFOCOM 2005*, Miami, FL, March 2005.

[30] Haerri, J., F. Filali and C. Bonnet. 2009. Mobility models for vehicular ad hoc networks: a survey and taxonomy. *IEEE Communications Surveys and Tutorials*, to appear.

[31] The Network Simulator, ns-2 http://www.isi.edu/nsnam/ns.

[32] GloMoSim, Global Mobile Information Systems Simulation Library, http://pcl.cs.ucla.edu/projects/glomosim

[33] QualNet network simulator, http://www.scalable-networks.com.

[34] OPNET Network Modeling and Simulation, http://www.opnet.com/services/university.

[35] BonnMotion, a mobility scenario generation and analysis tool, http://web.informatik.uni-bonn.de/IV/BonnMotion.

[36] U.S. Census Bureau, Topologically Integrated Geographic Encoding and Referencing (TIGER) system, http://www.census.gov/geo/www/tiger.

[37] Ertico, http://www.ertico.com/en/links/links/gdf_-_geographic_data_files.htm.

[38] Jardosh, A., E. Belding-Royer, K. Almeroth and S. Suri. 2003. Toward realistic mobility models for mobile ad hoc networks. In *9th Annual International Conference on Mobile Computing and Networking (MobiCom 2003)*, San Diego, CA, September 2003.

[39] Tian, J., J. Haehner, C. Becker, I. Stepanov and K. Rothermel. 2002. GraphBased mobility model for mobile ad-hoc network simulation. In *ACM Simulation Symposium*, 2002.

[40] Bettstetter, C. Smooth is better than sharp: a random mobility model for simulation of wireless networks. In *4th ACM International Work-shop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM 2001)*, Rome, Italy, July 2001.

[41] Seskar, I., S. Marie, J. Holtzman and J. Wasserman. 1992. Rate of location area updates in cellular systems. In *IEEE Vehicular Technology Conference (VTC'92)*, Denver, CO, May 1992.

[42] Trieber, M., A. Hennecke and D. Helbing. 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2): 1805–1824.

[43] Treiber, M. and D. Helbing. 2002. Realistische Mikrosimulation von Strassenverkehr mit einem einfachen Modell. In *16th Symposium Simulationstechnik (ASIM 2002)*, Rostock, September 2002.

[44] Karnadi, F., Z. Mo and K.-C. Lan. 2005. Rapid generation of realistic mobility models for VANET. In *Poster Session at the 11th Annual International Conference on Mobile Computing and Networking (MobiCom 2005)*, Cologne, Germany, August 2005.

[45] SUMO Simulation of Urban MObility, http://sumo.sourceforge.net.

[46] Schünemann, B., et al., eWorld—a Traffic Simulation Connector, Hasso Plattner Institute in Potsdam, Germany, http://eworld.sourceforge.net/.

[47] OpenStreetMap database (OSM), http://www.openstreetmap.org/.

[48] Choffnes, D., and F. Bustamante. 2005. An integrated mobility and traffic model for vehicular wireless networks. In *2nd ACM Workshop on Vehicular Ad Hoc Networks (VANET 2005)*, Cologne, Germany, September 2005.

[49] Anda, J., J. LeBrun, D. Ghosal, C.-N. Chuah and H. M. Zhang. 2005. VGrid: vehicular ad hoc networking and computing grid for intelligent traffic control. In *IEEE Vehicular Technology Conference (VTC Spring 2005)*, Stockholm, Sweden, May 2005.

[50] Mangharam, R., D. Weller, D. Stancil, R. Rajkumar and J. Parikh. 2005. GrooveSim: a topography-accurate simulator for geographic routing in vehicular networks. In *2nd ACM Workshop on Vehicular Ad Hoc Networks (VANET 2005)*, Cologne, Germany, September 2005.

[51] Jaap, S., M. Bechler and L. Wolf. 2005. Evaluation of routing protocols for vehicular ad hoc networks in city traffic scenarios. In *5th International Conference on Intelligent Transportation Systems Telecommunications (ITST)*, June 2005.

[52] Mahajan, A. et al. 2005. *Evaluation of Mobility Models for Vehicular Ad-hoc Network Simulations*. Technical Report N.051220, Florida State University, 2005.

[53] Uchiyama, A. 2005. Mobile ad-hoc network simulator based on realistic behavior model. In *6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2005)*, Urbana-Champaign, IL, May 2005.

[54] Bononi, L., M. Di Felice, M. Bertini and E. Croci. 2006. Parallel and distributed simulation of wireless vehicular ad hoc networks. In *ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2006)*, Torresmolinos, Spain, 2006.

[55] ARTiS: Advanced Runtime Infrastructure, http://pads.cs.unibo.it/dokuwiki/doku.php?id=pads:artis-doc.

[56] GPS TrackMaker, http://www.gpstm.com.

[57] Gorgorin, C., V. Gradinescu, R. Diaconescu, V. Cristea and L. Ifode. 2006. An integrated vehicular and network simulator for vehicular ad-hoc networks. In *European Simulation and Modelling Conference (ESM)*, Bonn, Germany, May 2006.

[58] Wang, S.Y., C.L. Chou, Y.H. Chiu, Y.S. Tseng, M.S. Hsu, Y.W. Cheng, W.L. Liu and T.W. Ho. 2007. NCTUns 4.0: an integrated simulation platform for vehicular traffic, communication, and network researches. In *1st IEEE International Symposium on Wireless Vehicular Communications (WiVec'07)*, Baltimore, MD, October 2007.

[59] Vuyyuru, R., and K. Oguchi. 2007. Vehicle to vehicle ad hoc communication protocol evaluation using Realistic Evaluation

Framework. In *4th Annual Conference on Wireless On demand Network Systems and Services (Wons'07)*, Obergurgl, Austria, January 2007.

[60] The SHIFT Traffic Simulator, http://path.berkeley.edu.

[61] UDel Models For Simulation of Urban Mobile Wireless Networks, http://udelmodels.eecis.udel.edu.

[62] Piorkowski, M., M. Raya, A. Lugo, P. Papdimitratos, M. Grossglauser and J. Hubaux. TraNS: Realistic Joint Traffic and Network Simulator for VANETS. In *13th Annual International Conference on Mobile Computing and Networking (MobiCom 2007), Student Poster session*, Montreal, QC, Canada, September 2007.

[63] Lochert, C., M. Caliskan, B. Scheuermann, A. Barthels, A. Cervantes and M. Mauve. 2005. Multiple simulator interlinking environment for inter vehicle communication. In *2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2005)*, Cologne, Germany, September 2005.

***Jérôme Härri** received a MSc degree and a Dr. ès sc. degree in telecommunication from the Swiss Institute Technology (EPFL), Lausanne, Switzerland. He is an assistant professor at the Institute of Telematics at Universität Karlsruhe (TH), Germany and leading the junior research group 'Traffic Telematics'. Previously, he was a research assistant and PhD student at EURECOM in Sophia-Antipolis, France working on mobility modeling and management for mobile wireless ad hoc networks. His research interests include inter-vehicular communication, vehicular mobility modeling, and intelligent transportation systems (ITSs; see http://nvt.tm.uni-karlsruhe.de for recent publications). He is currently involved into the activities of the CAR 2 CAR Communication Consortium in the working groups Communication and Simulation.*

***Marco Fiore** received a MS degree in computer science from the University of Illinois, Chicago, in 2003, and MSc and PhD degrees from Politecnico di Torino in 2004 and 2008, respectively. Between 2006 and 2007 he was a visiting doctoral student at Rice University, working with Professor Edward Knightly, and in 2008 he was a visiting researcher at Universitat Politècnica de Catalunya. He is currently a postdoctoral fellow at Politecnico di Torino. His research interests include mobility modeling and support, information sharing, and security in vehicular networks. He is a member of the IEEE.*

***Fethi Filali** received his PhD in Computer Science from the University of Nice Sophia-Antipolis in November 2002. He joined in September 2003 EURECOM as an Assistant Professor. His current research interests include the design, development, and performance evaluation of communication protocols, systems, and services for ubiquitous networking, communications for ITS, WIMAX, WSNs, DTNs, and WMNs. In April 2008, he was awarded the Habilitation from the University of Nice Sophia-Antipolis for his extensive research on wireless networking.*

***Christian Bonnet** joined EURECOM as an associate professor in 1992. Since 1998 he has been at the head of the Mobile Communications Department of EURECOM. His teaching activities are real-time and distributed systems, mobile communication systems, wireless LANs and protocols for mobility management. His main areas of research are wireless protocols, wireless access to IP networks and data communications in mobile networks including mobile ad hoc networks. He is currently participating in research projects related to UMTS: (IST) Moby Dick, (RNRT) SAMU, PLATON, @IRS++. Before joining Eurecom he was a consultant in the GSI group for 9 years where he worked on different projects related to radio telecommunications, value added networks, and real-time networks. He participated in the European projects (ESPRIT) DESCARTES, DRAGON, and REX. He was appointed Director of the Real Time Department of GSI Tecsi. He was born in Paris on 17 September 1955. He received an engineering degree from Ecole Nationale des Mines de Nancy in 1978. He first worked for ALSYS where he participated in a compiler production project for ADA.*