WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
*Wirel. Commun. Mob. Comput.* (2009)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.859

# A survey and comparative study of simulators for vehicular *ad hoc* networks (VANETs)

Francisco J. Martinez[1]*,†, Chai Keong Toh[2], Juan-Carlos Cano[3], Carlos T. Calafate[3] and Pietro Manzoni[3]

[1] *Computer and Systems Engineering Department, University of Zaragoza, Ciudad Escolar, Teruel, Spain*
[2] *Department of EEE, University of Hong Kong, Pokfulam Road, Hong Kong*
[3] *Computer Engineering Department, Technical University of Valencia, Campus de Vera, Spain*

## Summary

Wireless communication technologies have now greatly impact our daily lives. From indoor wireless LANs to outdoor cellular mobile networks, wireless technologies have benefited billions of users around the globe. The era of vehicular *ad hoc* networks (VANETs) is now evolving, gaining attention and momentum. Researchers and developers have built VANET simulation software to allow the study and evaluation of various media access, routing, and emergency warning protocols. VANET simulation is fundamentally different from MANETs (mobile *ad hoc* networks) simulation because in VANETs, vehicular environment imposes new issues and requirements, such as constrained road topology, multi-path fading and roadside obstacles, traffic flow models, trip models, varying vehicular speed and mobility, traffic lights, traffic congestion, drivers' behavior, etc. Currently, there are VANET mobility generators, network simulators, and VANET simulators. This paper presents a comprehensive study and comparisons of the various publicly available VANET simulation software and their components. In particular, we contrast their software characteristics, graphical user interface (GUI), popularity, ease of use, input requirements, output visualization capability, accuracy of simulation, etc. Finally, while each of the studied simulators provides a good simulation environment for VANETs, refinements and further contributions are needed before they can be widely used by the research community. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: vehicular *ad hoc* networks; simulation; wireless

## 1. Introduction

Mobile *ad hoc* networks (MANETs) are a type of wireless networks that do not require any fixed infrastructure. MANETs are attractive for situations where communication is required but deploying a fixed infrastructure is impossible.

Vehicular *ad hoc* networks (VANETs) represent a rapidly emerging research field and are considered essential for cooperative driving among vehicles on the road. VANETs are characterized by: (a) trajectory-based movements with prediction locations and time-varying topology, (b) varying number of vehicles with independent or correlated speeds, (c) fast

*Correspondence to: Francisco J. Martinez, Computer and Systems Engineering Department, University of Zaragoza, Ciudad Escolar, Teruel, Spain.

†E-mail: f.martinez@unizar.es

time-varying channel (e.g., signal transmissions can be blocked by buildings), (d) lane-constrained mobility patterns (e.g., frequent topology partitioning due to high mobility), and (e) reduced power consumption requirements. So far, the development of VANETs is backed by strong economical interests since vehicle-to-vehicle (V2V) communication allows the sharing of wireless channels for collision avoidance (improving traffic safety), improved route planning, and better control of traffic congestion [1].

Deploying and testing VANETs involves high cost and intensive labor. Hence, simulation is a useful alternative prior to actual implementation. Simulations of VANET often involve large and heterogeneous scenarios. Compared to MANETs, when we simulate VANETs, we must account for some specific characteristics found in a vehicular environment. Based on previous studies of mobility behavior of mobile users [2], existing models try to closely represent the movement patterns of users. Moreover, it is well known that mobility models can significantly affect simulation results. For results to be useful, it is important that the simulated model is as close to reality as possible [3]. For MANETs, the random waypoint model (RWP) is by far the most popular mobility model [4], but in a vehicular network, nodes (vehicles) can only move along streets, prompting the need for a road model. Another important aspect in VANETs is that nodes do not move independently of each other; they move according to well-established vehicular traffic models, so the results for MANETs may not be directly applicable. Moreover, the speed of these nodes are different (in MANETs, nodes' speed ranges from 0 to 5 m/s, while in VANETs speed ranges from 0 to 40 m/s).

The above motivates us to make a survey of existing VANET simulators. This paper is organized as follows. Section 2 presents an overview of existing simulators for VANETs. Section 3 discusses the various VANET mobility generators available. Section 4 discusses the various network simulators, their graphical interfaces and characteristics and provides a comparison. Section 5 presents the characteristics and comparisons of existing VANET simulators. Finally, Section 6 concludes this paper.

## 2. Overview of Simulators for VANETs

In this section, we review various publicly available VANET simulators that are currently in use by the research community. In our study, we exclude proprietary VANET mobility generators or network simulators, such as TSIS-CORSIM [5], Paramics [6], Daimler-Chrysler Farsi and Videlio, Carisma [7], VISSIM [8], QualNet [9], or OPNET [10]. We focus on freeware and open source tools that allow free access to simulator source code.

Figure 1 presents the taxonomy of VANET simulation software. We have classified existing VANET simulation software into three different categories. They are (a) vehicular mobility generators, (b) network simulators, and (c) VANET simulators.

Vehicular mobility generators are needed to increase the level of realism in VANET simulations. They generate realistic vehicular mobility traces to be used as an input for a network simulator. The inputs of the mobility generator include the road model, scenario parameters (i.e., maximum vehicular speed, rates of

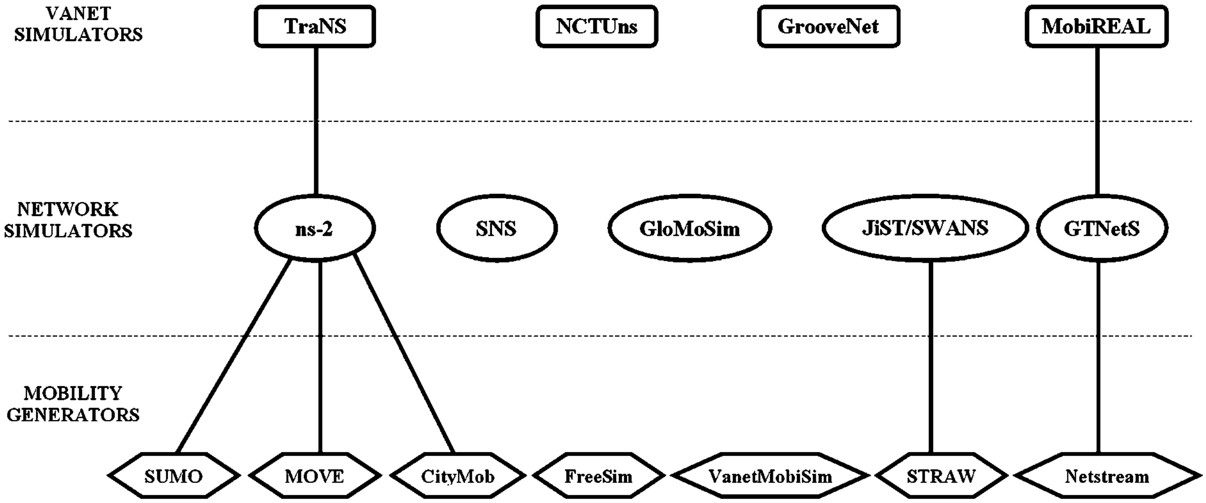


Fig. 1. A taxonomy of VANET simulation software.

DOI: 10.1002/wcm

vehicle arrivals and departures, etc). The output of the trace details the location of each vehicle at every time instant for the entire simulation time and their mobility profiles. Examples are SUMO [11], MOVE[‡] [12], CityMob [13], STRAW [14], FreeSim [15], Netstream [16], and VanetMobiSim [17].

Network simulators perform detailed packet-level simulation of source, destinations, data traffic transmission, reception, background load, route, links, and channels. Examples are ns-2 [18], GloMoSim [19], SNS [20], JiST/SWANS [21], and GTNetS [22]. Most existing network simulators are developed for MANETs and hence require VANET extensions (such as using the vehicular mobility generators) before they can be used to simulate vehicular networks.

Finally, VANET simulators provide both traffic flow simulation and network simulation. Examples are TraNS [23], NCTUns [24], GrooveNet [25], and MobiREAL [26]. In the next few sections, we will discuss in greater depth the functions, characteristics, and comparisons of vehicular mobility generators, network simulators, and VANET simulators.

## 3. VANET Mobility Generators

Vehicular mobility generators are needed to increase the level of realism in VANET simulations. In this section, we present the different vehicular traffic models, the existing mobility generators and a comparison of them.

### 3.1. Vehicular Traffic Models

Traffic modeling is a well-known research area in Civil Engineering and it is crucial to correctly model vehicular traffic during the design phase of new roads and intersections [27]. Transportation and traffic science classifies traffic models into macroscopic, mesoscopic, and microscopic models, according to the granularity with which traffic flows are examined. Macroscopic models, like METACOR [28], model traffic at a large scale, treating traffic like a liquid applying hydrodynamic flow theory to vehicle behavior. The simulation in a macroscopic model takes place on a section-by-section basis rather than by tracking individual vehicles. Macroscopic models have considerably fewer demands on computer requirements than microscopic models. However, they do not have the ability to analyze transportation improvements in as much detail as the microscopic models.

Mesoscopic models, such as *Continuous Traffic Assignment Model* (CONTRAM) [29], combine the properties of both microscopic and macroscopic simulation models. As in microscopic models, the mesoscopic models' unit of traffic flow is the individual vehicle. However, their movements follow the approach of the macroscopic model and is governed by the average speed on the travel link, so movements do not consider individual dynamic vehicle speed and volume relationships.

Simulations of VANET scenarios are concerned with the accurate modeling of radio wave transmissions among nodes and, therefore, require exact positions of simulated nodes. Since both macroscopic and mesoscopic models cannot offer this level of detail, microscopic simulations, which model the behavior of single vehicles and the interactions between them, are the most appropriate mobility models for simulating VANETs. Transportation and traffic science has developed a number of microsimulation models, each taking a dedicated approach ranging from coarse to fine grain. Models that have been widely used within the traffic science community include the *Cellular Automaton* (CA) model [30], the *Stefan Krauss* (SK) model [31], and the *Intelligent Driving Model* (IDM) [32]. Simulation time and memory requirements for microscopic models are high, usually limiting the network size and the number of simulation runs.

When dealing with vehicular mobility modeling, some authors [33] have distinguished between macro-mobility and micro-mobility. For macro-mobility, they refer to all the macroscopic aspects which influence vehicular traffic, i.e., the road topology, constrained car movements, the per-road speed limits, number of lanes, overtaking and safety rules for each street, or the traffic signs description establishing the intersections crossing rules. Micro-mobility refers instead to the drivers' individual behavior when interacting with other drivers or with the road infrastructure, i.e., traveling speed under different traffic conditions; acceleration, deceleration and overtaking criteria; behavior in the presence of road intersections and traffic signs, general driving attitude related to drivers' age, sex or mood, etc. It would be desirable for a trustworthy VANET simulation that both macro-mobility and micro-mobility descriptions are jointly considered when modeling vehicular movements.

[‡] MOVE is used to convert SUMO traffic traces into ns-2 or GloMoSim compatible traces.

DOI: 10.1002/wcm

### 3.2. Existing Mobility Generators

Nowadays, several simulation software environments exist and they are capable of generating trace files reflecting vehicles' movements.

- *VanetMobiSim* [17] is an extension of the CANU Mobility Simulation Environment (CanuMobiSim) [34] which focuses on vehicular mobility, and features realistic automotive motion models at both macroscopic and microscopic levels. At the macroscopic level, VanetMobiSim can import maps from the US Census Bureau topologically integrated geographic encoding and referencing (TIGER) database, or randomly generate them using Voronoi tesselation. The TIGER/Line files constitute a digital database of geographic features, such as roads, railroads, rivers, lakes, and legal boundaries, covering the entire United States. VanetMobiSim adds support for multi-lane roads, separate directional flows, differentiated speed constraints and traffic signs at intersections. At the microscopic level, it supports mobility models such as Intelligent Driving Model with Intersection Management (IDM/IM), Intelligent Driving Model with Lane Changing (IDM/LC) and an overtaking model (MOBIL), which interacts with IDM/IM to manage lane changes and vehicle accelerations and decelerations, providing realistic car-to-car and car-to-infrastructure interactions. VanetMobiSim is based on JAVA and can generate movement traces in different formats, supporting different simulation or emulation tools for mobile networks including ns-2 [18], GloMoSim [19], and QualNet [9].
- *SUMO* (Simulation of Urban MObility) [11] is an open source, highly portable, microscopic road traffic simulation package designed to handle large road networks. Its main features include collision free vehicle movement, different vehicle types, single-vehicle routing, multi-lane streets with lane changing, junction-based right-of-way rules, hierarchy of junction types, an openGL graphical user interface (GUI), and dynamic routing. SUMO can manage large environments, i.e., 10 000 streets, and it can import many network formats such as Visum [35], Vissim [8], ArcView [36], or XML-Descriptions. Thus, by combining SUMO and openstreetmap.org [37], we can simulate traffic in different locations of the globe. However, since SUMO is a pure traffic generator, its generated traces cannot be directly used by the available network simulators, which is a serious shortcoming.
- *MOVE* (MObility model generator for VEhicular networks) [12] rapidly generates realistic mobility models for VANET simulations. MOVE is built on top of SUMO. The output of MOVE is a mobility trace file that contains information of realistic vehicle movements which can be immediately used by popular network simulation tools such as ns-2 or GloMoSim. In addition, MOVE provides a GUI that allows the user to quickly generate realistic simulation scenarios without the hassle of writing simulation scripts as well as learning about the internal details of the simulator.
- *STRAW* (STreet RAndom Waypoint) [14] provides accurate simulation results by using a vehicular mobility model on real US cities, based on the operation of real vehicular traffic. STRAW's current implementation is written for the JiST/SWANS discrete-event simulator, and its mobility traces cannot be directly used by other network simulators, such as ns-2. STRAW is part of the C3 (Car-to-Car Cooperation) project [38]. A more realistic mobility model with the appropriate level of detail for vehicular networks is critical for accurate network simulation. The STRAW mobility model constrains node movement to streets defined by map data for real US cities and limits their mobility according to vehicular congestion and simplified traffic control mechanisms.
- *FreeSim* [15] is a fully customizable macroscopic and microscopic free-flow traffic simulator that allows for multiple freeway systems to be easily represented and loaded into the simulator as a graph data structure with edge weights determined by the current speeds. Traffic and graph algorithms can be created and executed for the entire network or for individual vehicles or nodes, and the traffic data used by the simulator can be user generated or be converted from real-time data gathered by a transportation organization. Vehicles in FreeSim can communicate with the system monitoring the traffic on the freeways, which makes FreeSim ideal for *Intelligent Transportation System* (ITS) simulation. FreeSim is licensed under the GNU General Public License, and the source code is available freely for download.
- *CityMob v.2* [13] CityMob is a ns-2 compatible mobility model generator proposed for use in VANETs. Citymob implements three different mobility models: (a) *Simple Model* (SM), (b) *Manhattan Model* (MM), and (c) realistic *Downtown Model* (DM). In DM model, streets are arranged in a Manhattan style grid, with a uniform block size

across the simulation area. All streets are two-way, with lanes in both directions. Car movements are constrained by these lanes. Vehicles will move with a random speed, within an user-defined range of values. DM model also simulates semaphores at random positions (not only at crossings), and with different delays. DM adds traffic density in a way similar to a real town, where traffic is not uniformly distributed. Hence, there will be zones with a higher vehicle density. These zones are usually in the downtown, and vehicles must move more slowly than those in the outskirts.

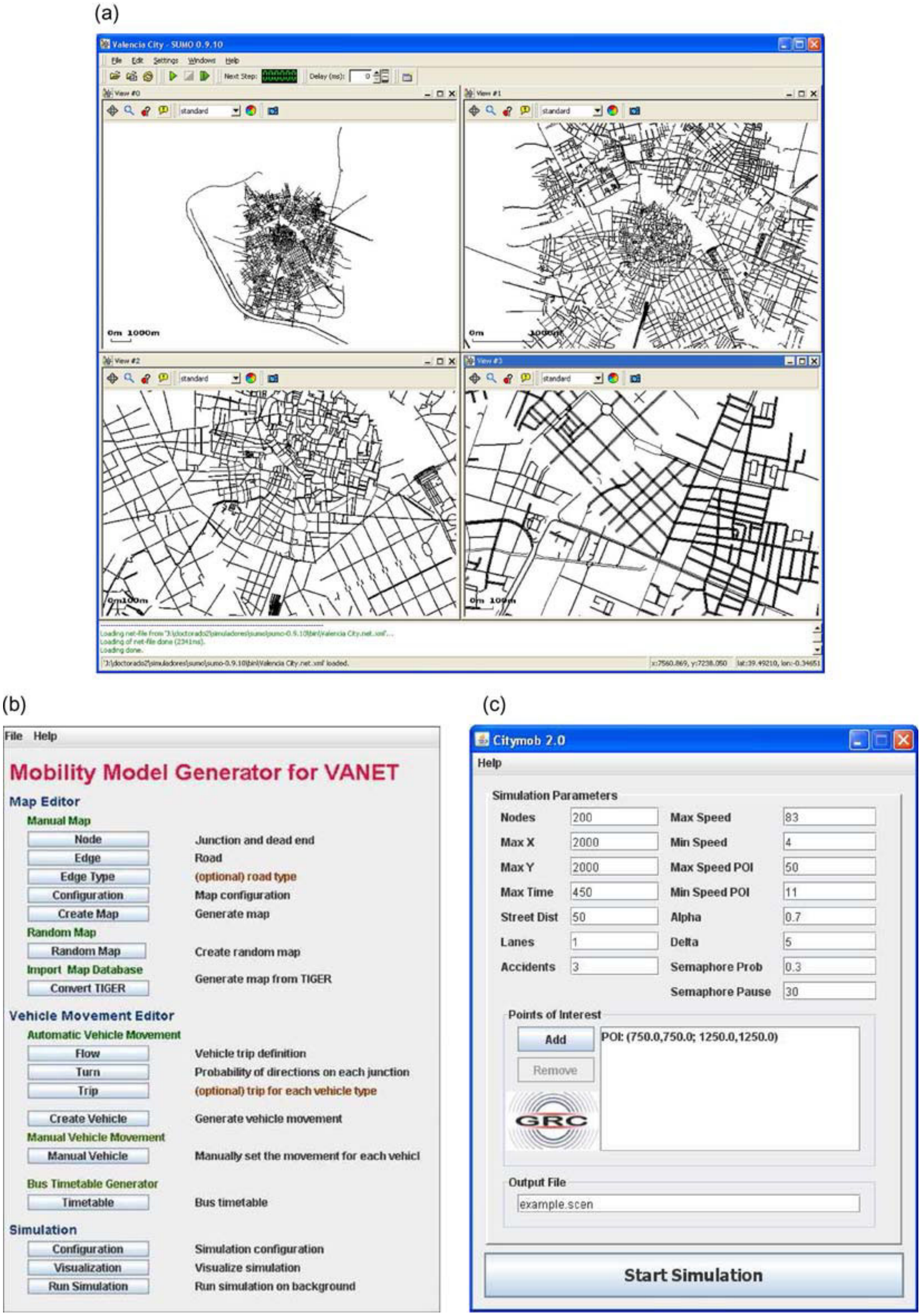
CityMob DM also has the following capabilities: (a) multiple lanes in both directions for every street, (b) vehicle queues due to traffic jams, and (c) the possibility of having more than a downtown.

### 3.3. Qualitative Comparison of Mobility Generators

To conclude this section, Table I presents a summary of the studied vehicular mobility generators focusing on their main characteristics. We have grouped the comparisons into five different categories: (a) software characteristics, (b) maps types, (c) mobility models supported, (d) traffic models implemented, and (e) trace formats supported.

Table I. A comparison of the studied mobility generators.

| | VanetMobiSim | SUMO | MOVE | STRAW | FreeSim | CityMob |
|---|---|---|---|---|---|---|
| Software | | | | | | |
| Portability | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Freeware | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Opensource | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Console | × | ✓ | ✓ | — | × | ✓ |
| GUI | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Available examples | ✓ | ✓ | ✓ | — | ✓ | × |
| Continuous development | × | ✓ | × | × | — | ✓ |
| Ease of setup | Moderate | Moderate | Easy | Moderate | Easy | Easy |
| Ease of use | Moderate | Hard | Moderate | Moderate | Easy | Easy |
| Maps | | | | | | |
| Real | ✓ | ✓ | ✓ | ✓ | ✓ | × |
| User defined | ✓ | ✓ | ✓ | — | × | × |
| Random | ✓ | ✓ | ✓ | × | × | ✓ |
| Manhattan | × | × | × | × | × | ✓ |
| Voronoi | ✓ | × | × | × | × | × |
| Mobility | | | | | | |
| Random waypoint | ✓ | ✓ | ✓ | × | × | ✓ |
| STRAW | × | ✓ | ✓ | ✓ | × | × |
| Manhattan | × | ✓ | ✓ | × | × | ✓ |
| Downtown | × | × | × | × | × | ✓ |
| Traffic models | | | | | | |
| Macroscopic | × | × | × | × | ✓ | × |
| Microscopic | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Multilane roads | ✓ | ✓ | ✓ | ✓ | — | ✓ |
| Lane changing | ✓ | ✓ | ✓ | ✓ | — | ✓ |
| Separate directional flows | ✓ | ✓ | ✓ | ✓ | — | ✓ |
| Speed constraints | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Traffic signs | ✓ | ✓ | ✓ | ✓ | — | ✓ |
| Intersections management | ✓ | ✓ | ✓ | — | — | × |
| Overtaking criteria | ✓ | — | — | — | — | × |
| Large road networks | — | ✓ | ✓ | ✓ | — | ✓ |
| Collision free movement | — | ✓ | ✓ | — | — | ✓ |
| Different vehicle types | × | ✓ | ✓ | — | × | ✓ |
| Hierarchy of junction types | × | ✓ | ✓ | — | × | × |
| Route calculation | ✓ | ✓ | ✓ | ✓ | ✓ | × |
| Traces | | | | | | |
| ns-2 trace support | ✓ | × | ✓ | × | × | ✓ |
| GloMoSim support | ✓ | × | ✓ | × | × | × |
| QualNet support | ✓ | × | ✓ | × | × | × |
| SWANS support | × | × | × | ✓ | × | × |
| XML-based trace support | ✓ | × | × | × | × | × |
| Import different formats | ✓ | ✓ | ✓ | × | × | × |



```
Loading net-file from "J:\doctorado2\simuladores\sumo\sumo-0.9.10\bin\Valencia City.net.xml"...
Loading of net-file done (2341ms).
Loading done.
"J:\doctorado2\simuladores\sumo\sumo-0.9.10\bin\Valencia City.net.xml" loaded.
x:7560.869, y:7238.050   lat:39.49210, lon:-0.34651
```

Mobility Model Generator for VANET

Map Editor

Manual Map

| | |
|---|---|
| Node | Junction and dead end |
| Edge | Road |
| Edge Type | (optional) road type |
| Configuration | Map configuration |
| Create Map | Generate map |

Random Map

| | |
|---|---|
| Random Map | Create random map |

Import Map Database

| | |
|---|---|
| Convert TIGER | Generate map from TIGER |

Vehicle Movement Editor

Automatic Vehicle Movement

| | |
|---|---|
| Flow | Vehicle trip definition |
| Turn | Probability of directions on each junction |
| Trip | (optional) trip for each vehicle type |
| Create Vehicle | Generate vehicle movement |

Manual Vehicle Movement

| | |
|---|---|
| Manual Vehicle | Manually set the movement for each vehicl |

Bus Timetable Generator

| | |
|---|---|
| Timetable | Bus timetable |

Simulation

| | |
|---|---|
| Configuration | Simulation configuration |
| Visualization | Visualize simulation |
| Run Simulation | Run simulation on background |

Simulation Parameters

| | | | |
|---|---|---|---|
| Nodes | 200 | Max Speed | 83 |
| Max X | 2000 | Min Speed | 4 |
| Max Y | 2000 | Max Speed POI | 50 |
| Max Time | 450 | Min Speed POI | 11 |
| Street Dist | 50 | Alpha | 0.7 |
| Lanes | 1 | Delta | 5 |
| Accidents | 3 | Semaphore Prob | 0.3 |
| | | Semaphore Pause | 30 |

Points of Interest

POI: (750.0,750.0; 1250.0,1250.0)

Output File

example.scen

Fig. 2. Graphical user interfaces (GUIs) of (a) SUMO, (b) MOVE, and (c) CityMob v.2.

DOI: 10.1002/wcm

As shown, there is no mobility generator that can fulfill all the desired capabilities needed by the research community. Freesim exhibits good software characteristics but is limited in other functions. CityMob is good in software features and traffic model support. SUMO, MOVE, STRAW, and VanetMobiSim all have good software features and traffic model support. However, only VanetMobiSim provides excellent trace support.

As for GUIs, they are intuitive and user friendly. Figure 2 shows the GUIs of some VANET mobility generators.

### 3.4. Performance Comparison of Mobility Generators

In most VANET simulations, researchers evaluate the effectiveness and performance of their proposed *Warning Message Dissemination* (WMD) protocol. Such a protocol is viewed to be useful when an accident occurs. It can help to alleviate congestion and warn other vehicles of the accident. To evaluate the realism and effectiveness of existing VANET mobility generators, we performed simulation of a generic WMD protocol on ns-2 over VanetMobiSim, CityMob, SUMO, and real traces. Real trace was obtained from real mobility, as in the Cabspotting project [39]. Table II shows the simulation parameters used. Note that we have included an 802.11p implementation into the ns-2.

The performance parameters we measured include: (a) the percentage of blind vehicles, (b) the warning message notification time, and (c) the number of packets received per vehicle. The percentage of blind vehicles is the percentage of vehicles that do not receive the warning messages sent by the accident vehicle. The warning notification time is the time required by normal vehicles to receive the warning message.

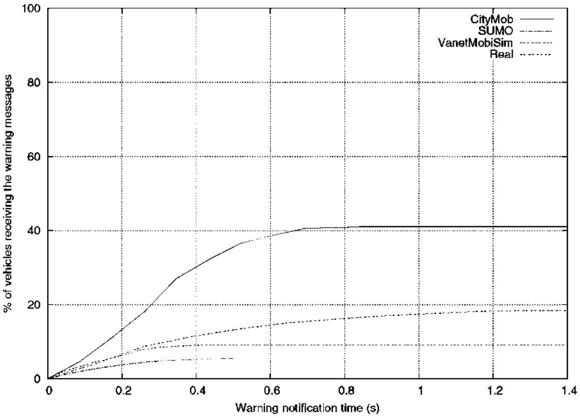


Fig. 3. Cumulative histogram for the time evolution of disseminated warning messages using different mobility traces.

As shown in Figure 3, CityMob trace results in the shortest warning notification time, followed by Real, VanetMobiSim, and SUMO. In terms of percentage of blind vehicles, SUMO resulted in the largest percentage of blind vehicles (Figure 4a). Finally, in terms of number of packets received, CityMob outperforms all others, with VanetMobiSim and SUMO having similar results.

Our investigation shows that when simulating the same WMD protocol with the same network simulator

Table II. Parameters used for performance simulation of different VANET mobility generators.

| | | | | |
|---|---|---|---|---|
| Network simulator | ns-2.31 | | | |
| VANET mobility generator | Real trace | CityMob | SUMO | VanetMobiSim |
| Number of nodes[1] | 221 | | | |
| Map area size | 6500 m × 6500 m | | | |
| Maximum speed[2] | 40.23 km/h | | | |
| Distance between streets | — | 100 m | 100 m | — |
| Downtown size | — | 2000 m × 2000 m | — | — |
| Downtown speed (min–max) | — | 11–30 km/h | — | — |
| Downtown probability | — | 0.6 | — | — |
| Number of warning mode nodes | 3 | | | |
| Warning packet size | 256 B | | | |
| Normal packet size | 512 B | | | |
| Packets sent by nodes | 1 per second | | | |
| Warning message priority | AC3 | | | |
| Normal message priority | AC1 | | | |
| MAC/PHY | 802.11p | | | |
| Maximum transmission range | 250 m | | | |

[1] Real trace had this number of vehicles.

[2] The maximum speed allowed in San Francisco.

DOI: 10.1002/wcm

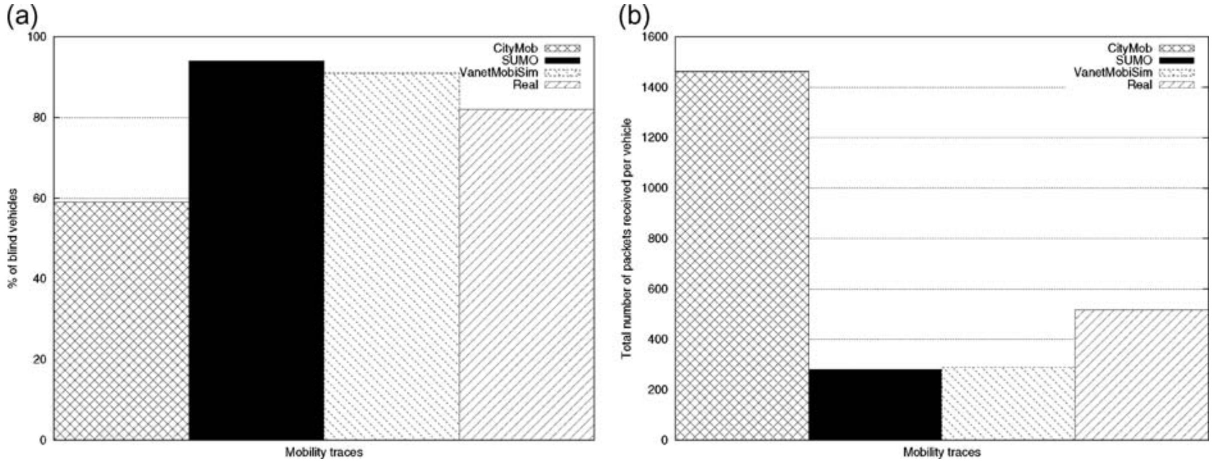


Fig. 4. Results for (a) percentage of blind vehicles and (b) total number of packets received at each vehicle for the different mobility traces.

over different VANET mobility generators, different performance results will be obtained. Such deviations can make results unconvincing and inconclusive. So far, only VanetMobiSim produces results in close resemblance to the Real trace.

## 4. Network Simulators

Network simulators allow researchers to study how the network would behave under different conditions. Users can then customize the simulator to fulfill their specific analysis needs. Compared to the cost and time involved in setting up an entire testbed containing multiple networked computers, routers and data links, network simulators are relatively fast and inexpensive. Hence, they allow researchers to test scenarios that might be particularly difficult or expensive to emulate using real hardware, especially in VANETs. Network simulators are particularly useful to test new networking protocols or to propose modifications to existing ones in a controlled and reproducible manner.

### 4.1. Existing Network Simulators

Several network simulators can be used to simulate the communication between vehicles in inter-vehicle communication (IVC) systems. Next we present the main characteristics of some of the most promising network tools to simulate VANET scenarios.

- *ns-2* [18] is a discrete event simulator developed by the VINT project research group at the University of California at Berkeley. The simulator was extended by the Monarch research group at Carnegie Mellon University [40] to include: (a) node mobility, (b) a realistic physical layer with a radio propagation model, (c) radio network interfaces, and (d) the IEEE 802.11 *Medium Access Control* (MAC) protocol using the distributed coordination function (DCF).

  However, the ns-2 distribution code had some significant shortcomings both in the overall architecture and the modeling details of the IEEE 802.11 MAC and PHY modules. In Reference [41], authors presented a completely revised architecture and design for these two modules. The resulting PHY is a full featured generic module capable of supporting any single channel frame based communications. The key features include cumulative signal to interference plus noise ratio (SINR) computation, preamble and physical layer convergence procedure (PLCP) header processing and capture, and frame body capture. The MAC now accurately models the basic IEEE 802.11 carrier sense multiple access with collision avoidance (CSMA/CA) mechanism, as required for credible simulation studies.
- *GloMoSim* [19] is a scalable simulation environment for wireless and wired network. It has been designed using the parallel discrete-event simulation capability provided by Parsec [42]. GloMoSim has been built using a layered approach similar to the OSI seven layer protocol model. Standard APIs are used between the different simulation layers. This allows the rapid integration of models developed at different layers by different people. The widely used

DOI: 10.1002/wcm

QualNet [9] simulator is a commercial version of GloMoSim.

- *JiST/SWANS* [21]. JiST is a high performance discrete event simulation engine that runs over a standard Java virtual machine. It is a prototype of a new general purpose approach to building discrete event simulators, that unifies the traditional systems and language-based simulator designs. It outperforms existing highly optimized simulation engines both in time and memory consumption. Simulation code that runs on JiST need not be written in a domain-specific language invented specifically for writing simulations, nor must it be littered with special purpose system calls and 'call backs' to support runtime simulation. Instead, JiST converts an existing virtual machine into a simulation platform, by embedding simulation time semantics at the byte-code level. Thus, JiST simulations are written in Java, compiled using a regular Java compiler, and run over a standard, unmodified virtual machine.

  SWANS is a scalable wireless network simulator built on top of the JiST platform. It was created primarily because existing network simulation tools are not sufficient for current research needs. SWANS contains independent software components that can be composed to form complete a wireless network or sensor network. Its capabilities are similar to ns-2 and GloMoSim, but SWANS is able of simulating much larger networks. SWANS leverages the JiST design to achieve higher simulation throughput, lower memory requirements, and run standard Java network applications over simulated networks. SWANS can simulate networks that are one or two orders of magnitude larger than what is possible with GloMoSim and ns-2, respectively, using the same amount of time and memory, and with a same level of detail [43].
- *SNS* (a Staged Network Simulator) [20]. Traditional wireless network simulators are limited in speed and scale because they perform many redundant computations both within a single simulation run, as well as across multiple invocations of the simulator. The staged simulation technique [44] proposes to eliminate redundant computations through function caching and reuse. The central idea behind staging is to cache the results of expensive operations and reuse them whenever possible. SNS is a staged simulator based on ns-2. On a commonly used *ad hoc* network simulation setup with 1500 nodes, SNS executes approximately 50 times faster than regular ns-2 and 30% of this improvement is due to staging, and the rest to engineering. This level of performance enables SNS to simulate large networks. However, the current implementation is based on ns-2 version 2.1b9a, and it is not specifically designed to simulate VANET scenarios.

## 4.2. Comparison of Network Simulators

In Table III, we present a summary of the studied network simulators and their characteristics. As shown, ns-2 is less suitable for simulating large networks but it is popular and easy to use, unlike SNS or JiST/SWAN. In fact, JiST/SWAN is the most difficult to install. All the studied simulators provide open source code and are available freely on the Internet. Users can modify

Table III. A comparison of the studied network simulators.

| | ns-2 | GloMoSim | JiST/SWANS | SNS |
|---|---|---|---|---|
| Software | | | | |
| Portability | ✓ | ✓ | ✓ | ✓ |
| Freeware | ✓ | ✓ | ✓ | ✓ |
| Opensource | ✓ | ✓ | ✓ | ✓ |
| Available examples | ✓ | ✓ | ✓ | ✓ |
| Continuous development | ✓ | × | ✓ | × |
| Large networks | × | ✓ | ✓ | ✓ |
| Console | ✓ | ✓ | ✓ | ✓ |
| GUI | ✓ | ✓ | ✓ | ✓ |
| Scalability | Poor | High | High | High |
| Ease of setup | Easy | Moderate | Hard | Easy |
| Ease of use | Hard | Hard | Hard | Hard |
| VANET | | | | |
| 802.11p | Only for ns-2.33 | × | × | × |
| Obstacles | × | × | × | × |
| Vehicular traffic flow model | × | × | × | × |

DOI: 10.1002/wcm

and enhance the code further, creating new versions. The major shortcoming is the lack of considerations for VANETs. For example, vehicular traffic flow models are not considered and 802.11p MAC is not included into the simulators (except for ns-2.33). Also, physical layer issues, obstacles, and road topologies present in a vehicular environment are often neglected.

## 5. VANET Simulators

One important aspect in a simulation model for an IVC system is the drivers' response to the IVC application. The reaction of drivers in different situations could affect traffic throughput [45]. For example, a driver who receives a collision warning message can either hit the brake or exit the highway, depending on the distance to the accident scene and the availability of exits.

The software that allows one to change the behavior of vehicles (depending on a given application context) is known as an integrated framework or simply a VANET simulator.

### 5.1. Existing VANET Simulators

To the best of our knowledge, there are only a few integrated frameworks available. Currently, the mobility and network models in integrated frameworks are implemented in two separated simulation tools. Therefore, there is a clear need for an integrated mobility and network simulator in order to evaluate effectively performance of IVC systems. Below, we discuss these simulators.

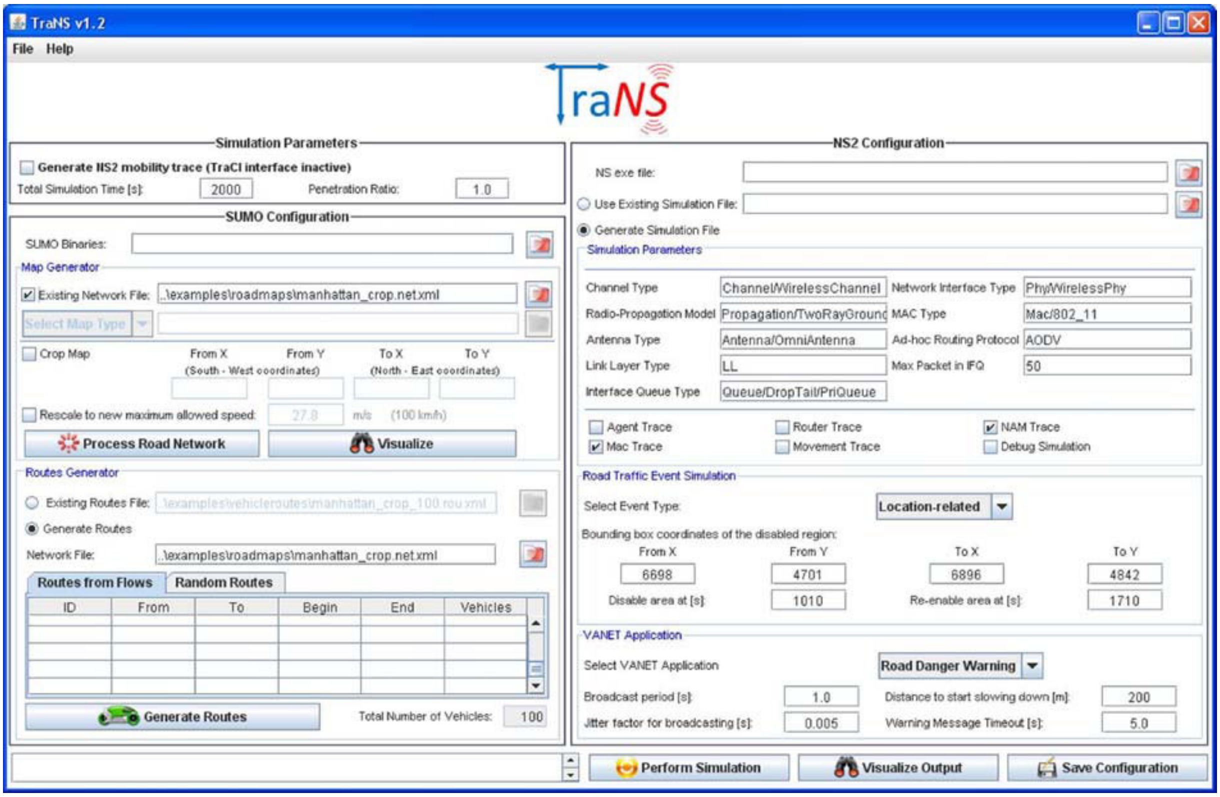
- *TraNS* (Traffic and Network Simulation Environment) [23] (see Figure 5) is a simulation environment that integrates both a mobility generator and a network simulator and it provides a tool to build realistic VANET simulations. TraNS provides a feedback between the vehicle behavior and the mobility model. For example, when a vehicle broadcasts information reporting an accident, some of the neighboring vehicles may slow down. TraNS is an open open-source project providing an application-centric evaluation framework for VANETs.

  TraNS is written in Java and C++ and works under Linux and Windows (trace-generation mode). The current implementation of TraNS uses the SUMO traffic simulator and the ns-2 network simulator. It is being developed at EPFL, Switzerland.

  TraNS v1.2 has several features, including: (a) support for realistic 802.11p, (b) automated generation of road networks from TIGER and Shapefile maps, (c) automated generation of random vehicle routes, (d) mobility trace generation for ns-2, SUMO and ns-2 coupling through the TraCI [46] interface, and (e) possibility to simulate road traffic events, e.g., accidents. Moreover, it provides two ready-to-use VANET applications: (a) Road Danger Warning (safety), and (b) Dynamic Reroute (traffic efficiency). TraNS can simulate large-scale networks (tested up to 3000 vehicles), and allows for Google Earth visualization of simulations (currently works for TIGER files only).
- *GrooveNet* [25] (see Figure 6a) is a hybrid simulator which enables communication between simulated vehicles and real vehicles. By modeling Inter-Vehicular Communication within a real street map based topography, it eases protocol design and in-vehicle deployment. GrooveNet's modular architecture incorporates mobility, trip and message broadcast models over a variety of link and physical layer communication models. GrooveNet supports simulations of thousands of vehicles in any US city as well as the addition of new models for networking, security, applications, and vehicular interactions. It provides multiple network interfaces, and allows GPS and event-triggered (from the vehicles' on-board computer) simulations.

  GrooveNet supports three types of simulated nodes: (a) vehicles which are capable of multi-hopping data over one or more dedicated short-range communications (DSRC) channels, (b) fixed infrastructure nodes, and (c) mobile gateways capable of V2V and *vehicle-to-infrastructure* (V2I) communication. GrooveNet supports multiple message types such as GPS messages, which are broadcast periodically to inform neighbors of a vehicle's current position, and vehicle emergency and warning event messages with priorities. Multiple rebroadcast policies have been implemented to investigate the broadcast storm problem. GrooveNet is able to support hybrid simulations where the simulated vehicle position, direction and messages are broadcast over the cellular interface from one or more infrastructure nodes. Real vehicles communicate only with those simulated vehicles which are within its transmission range. GrooveNet generates street level maps for any place in the USA by importing TIGER files which are available free from the US Census Bureau. GrooveNet is based on open-source roadnav [47] with significant additions, including a graph-based abstraction of streets, networking, simulation models, and a cross-platform GUI in Qt [48].

DOI: 10.1002/wcm



Simulation Parameters

☐ Generate NS2 mobility trace (TraCI interface inactive)

Total Simulation Time [s]: 2000
Penetration Ratio: 1.0

SUMO Configuration

SUMO Binaries:

Map Generator

☑ Existing Network File: ..\examples\roadmaps\manhattan_crop.net.xml

Select Map Type

☐ Crop Map
From X, From Y (South - West coordinates); To X, To Y (North - East coordinates)

☐ Rescale to new maximum allowed speed: 27.8 m/s (100 km/h)

Routes Generator

○ Existing Routes File: ..\examples\vehicleroutes\manhattan_crop_100.rou.xml

● Generate Routes

Network File: ..\examples\roadmaps\manhattan_crop.net.xml

| ID | From | To | Begin | End | Vehicles |
|---|---|---|---|---|---|
| | | | | | |

Total Number of Vehicles: 100

NS2 Configuration

NS exe file:

○ Use Existing Simulation File:

● Generate Simulation File

Simulation Parameters

| | | | |
|---|---|---|---|
| Channel Type | Channel/WirelessChannel | Network Interface Type | Phy/WirelessPhy |
| Radio-Propagation Model | Propagation/TwoRayGround | MAC Type | Mac/802_11 |
| Antenna Type | Antenna/OmniAntenna | Ad-hoc Routing Protocol | AODV |
| Link Layer Type | LL | Max Packet in IFQ | 50 |
| Interface Queue Type | Queue/DropTail/PriQueue | | |

☐ Agent Trace ☐ Router Trace ☑ NAM Trace
☑ Mac Trace ☐ Movement Trace ☐ Debug Simulation

Road Traffic Event Simulation

Select Event Type: Location-related

Bounding box coordinates of the disabled region:

| From X | From Y | To X | To Y |
|---|---|---|---|
| 6698 | 4701 | 6896 | 4842 |

Disable area at [s]: 1010
Re-enable area at [s]: 1710

VANET Application

Select VANET Application: Road Danger Warning

Broadcast period [s]: 1.0
Distance to start slowing down [m]: 200
Jitter factor for broadcasting [s]: 0.005
Warning Message Timeout [s]: 5.0

Fig. 5. Graphical interface of TraNS.

- *NCTUns* (National Chiao Tung University Network Simulator) [24] (see Figure 6b) is a high-fidelity and extensible network simulator and emulator capable of simulating various protocols used in both wired and wireless IP networks. Its core technology is based on a novel kernel re-entering methodology. Due to this novel methodology, NCTUns provides many unique advantages that cannot be easily achieved by traditional network simulators such as ns-2 and OPNET.

  The NCTUns network simulator and emulator has many useful features. It can be easily used as an emulator since it supports seamless integration of emulation and simulation. It uses Linux TCP/IP protocol stack to generate high-fidelity simulation results. It can run any real-life UNIX application program on a simulated node without any modifications. Supported networks include Ethernet-based fixed Internet, IEEE 802.11b wireless LANs, IEEE 802.11e QoS wireless LANs, IEEE 802.16d WiMAX wireless networks, DVB-RCS satellite networks, wireless vehicular networks for Intelligent Transportation Systems (including V2V and V2I), multi-interface mobile nodes for heterogeneous wireless networks, IEEE 802.16e mobile WiMAX networks, IEEE 802.11p/1609 WAVE wireless vehicular networks, etc.

  NCTUns supports parallel simulations on multi-core machines. By using an innovative parallel simulation approach, it supports parallel simulations for fixed networks on multi-core machines. It also provides a highly integrated and professional GUI environment that can help a user to quickly: (1) draw network topologies, (2) configure the protocol modules used inside a node, (3) specify the moving paths of mobile nodes, (4) plot network performance graphs, (5) play back the animation of a logged packet transfer trace, etc. All of these operations can be easily, intuitively, and quickly done with the GUI. Its main drawback is that NCTUns requires Fedora 9 Linux distribution to be installed, which poses a big problem for the majority of VANET researchers, limiting its wide usage.
- *MobiREAL* [26] (see Figure 6c) provides a new methodology to model and simulate realistic mobility of nodes and evaluate MANET applications. It is a network simulator that can simulate realistic mobility of humans and vehicles, and allow the changing

DOI: 10.1002/wcm

of their behavior depending on a given application context. MobiREAL can easily describe mobility of nodes using C++. It adopts a probabilistic rule-based model to describe the behavior of mobile nodes, which is often used in cognitive modeling of human behavior. The proposed model allows one to describe how mobile nodes can change their destinations, routes and speeds/directions based on their positions, surroundings (obstacles and neighboring nodes), and information obtained from applications.

MobiREAL simulates MANETs by using the mobility support found in the Georgia Tech Network Simulator (GTNetS) [22]. MobiREAL Animator dynamically visualizes node movement, connectivity states, and packet transmission. This enhances the understanding of simulation results intuitively. Mobility of nodes is simulated in the Behavior Simulator. Also an algorithm for collision avoidance among pedestrians is implemented. Traffic congestion of vehicles can also be modeled. Using MobiReal, it is possible to simulate a mixture of various mobility models concurrently.

For vehicular mobility, the authors had modified a traffic simulator called NETSTREAM [16] that was developed by TOYOTA. Since NETSTREAM is a proprietary software, its main drawback is that users cannot access and modify this part of the simulator, limiting its wide usage. However, MobiREAL can use other traffic simulators to provide vehicular mobility.

## 5.2. Comparison of VANET Simulators

In Table IV, we present a comparison of the studied VANET simulators. As shown, TraNS uses SUMO and ns-2 while MobiREAL uses GTNetS as the underlying network simulator. All simulators support different mobility models and provide microscopic traffic simulation. NCTUns provides random speed models while the others model street speed. Currently, all simulators support trip and intersection models. So far, only TraNS and NCTUns have an implementation of 802.11p and only GrooveNet and TraNS provide built-in VANET applications. In terms of ease of setup, NCTUns is viewed to be the hardest. In terms of ease of use, TraNS and GrooveNet are preferred.

Since these simulators were developed with different focus, results obtained when simulating similar VANET scenarios can differ greatly. TraNS and GrooveNet were developed to simulate VANETs. NCTUns was created for more general network

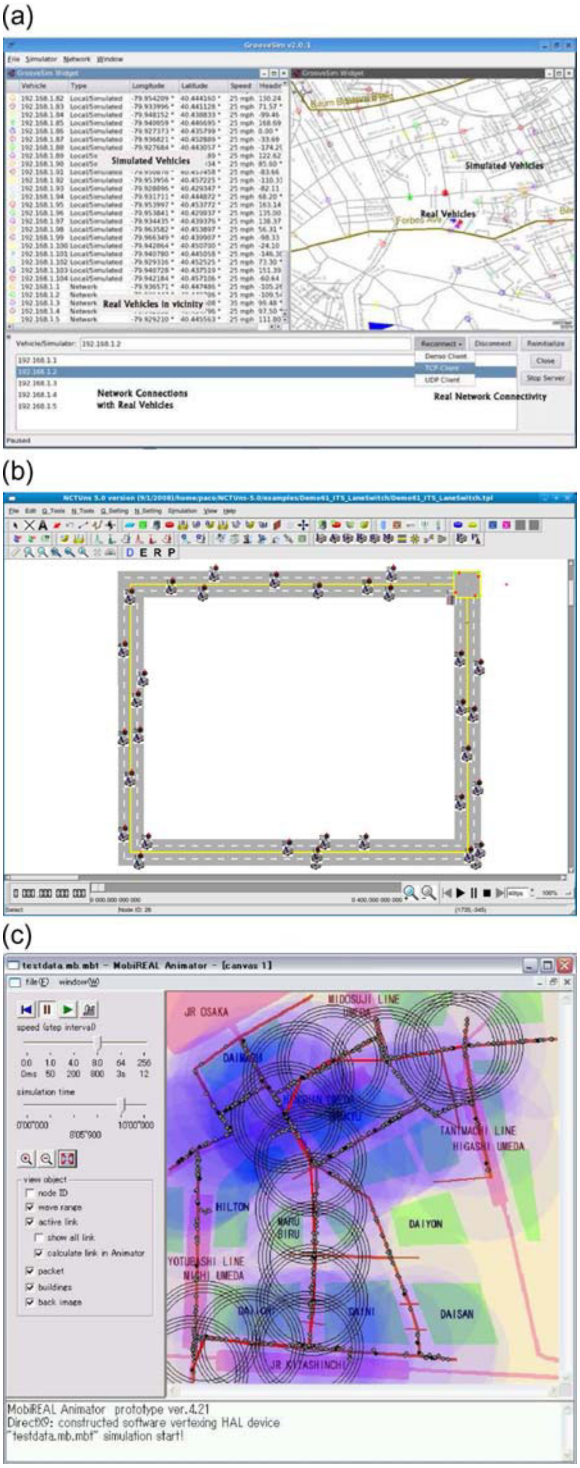


Fig. 6. Graphical user interfaces (GUIs) of (a) GrooveNet, (b) NCTUns, and (c) MobiREAL.

DOI: 10.1002/wcm

Table IV. A comparison of the studied VANET simulators.

| | TraNS | GrooveNet | NCTUns | MobiReal |
|---|---|---|---|---|
| Mobility generator | SUMO | GrooveNet | NCTUns | MobiReal |
| Network simulator | ns-2 | | | based on GTNetS |
| Mobility models | Random and manual routes | Random waypoint, explicit origin-destination, distributed origin-dest | Random and manual routes | probabilistic rule-based |
| Simulation type | Microscopic, space-continuous and time-discrete | | | |
| Lane models | Multi-lane streets with lane changing | | | |
| Speed models | Street speed | Uniform, street speed, Markov model, load-based | Random | Street speed |
| Traffic flow model | Car following SK and traffic assignment using the DUA-approach | Car following | Car following | Car following |
| Road topology | Any | Any | User defined | Any |
| Traffic lights | Manually defined | Manually defined | Automatically generated on intersections | Manually defined |
| Intersection model | Junction-based right-of-way rules<br>Hierarchy of junction types | Managed by traffic lights | Managed by four traffic lights | right-of-way rules and managed by traffic lights |
| Trip model | Random, manually defined | Djikstra, sightseeing | Manually defined | Manually defined |
| VANET protocols and facilities | 802.11p two ready-to-use VANET applications: road danger warning (safety) and dynamic reroute (traffic efficiency) tested up to 3000 vehicles | Supports V2V and V2I communications multiple message types, which are broadcast periodically to inform neighbors of a vehicle's current position, and vehicle emergency with priorities | 802.11p, supports multiple interfaces at the same time car agents control the driving behavior moving on a road. | Initially it was especially designed for MANETs instead of VANETs. |
| VANET built-in application support | Road danger warning and dynamic reroute | Vehicle warning and adaptive rebroadcast | None | None |
| Ease of setup | Moderate | Moderate | Hard | Easy |
| Ease of use | Moderate | Hard | Hard | Hard |
| Comments | Integrates both traffic and network simulators. Information exchanged in communication protocols can influence the vehicle behavior in the mobility model. | Able to support hybrid simulations (i.e., communication between simulated vehicles and real vehicles on the road) | Supports seamless integration of emulation and simulation, but it needs Fedora nine Operating System to be installed | Simulates realistic mobility of humans and cars, and their behavior can be changed depending on the given application context |

simulation purposes, while MobiREAL was designed for simulating MANETs. MobiREAL was recently enhanced to support VANET simulation.

Table V presents a comparison of the GUIs provided by the studied VANET simulators. All simulators provide both alphanumeric and config file input and console message output but the user interface for TraNS appears more sophisticated than others. A lot of manual parameter inputs are needed for TraNS. All simulators provide street-level topology view. So far, only TraNS can support visualization using Google Earth.

Finally, Table VI presents a comparison on the popularity of the studied VANET simulators. To obtain the popularity results, we used the IEEE Explorer [49], and the Google Scholar [50] tools to extract papers

DOI: 10.1002/wcm

Table V. A comparison of VANET GUIs.

| GUI | User friendly | Topology view | Parameters input | Output |
|---|---|---|---|---|
| TraNS | Good | Google Earth<br>With zoom ability<br>Without obstacles | Street map file<br>Mobility file<br>Graphical input | ns-2 trace<br>.kmz file (Google Earth) |
| GrooveNet | Good | Street view<br>With zoom ability<br>Without obstacles | Street map file<br>Simulation file<br>Graphical input | Simulation file<br>Animation view |
| NCTUns | Moderate | User defined<br>With zoom ability<br>With obstacles | Topology file<br>Graphical input | Simulation file<br>Animation view |
| MobiREAL | Moderate | User defined<br>With zoom ability<br>Without obstacles | Street map file<br>Mobility models<br>Density and routes file<br>Graphical input | Trace file<br>Animation view |

Table VI. Popularity comparison.

| | TraNS | GrooveNet | NCTUns | MobiREAL |
|---|---|---|---|---|
| Number of published papers that use the simulator | 2 | 8 | 13 | 2 |
| Number of citations found on IEEE xplore | 5 | 5 | 9 | 0 |
| Number of citations found on google scholar | 5 | 4 | 14 | 2 |

Data obtained on 20 January 2009.

published from the year 2000 to 2009. We consider only papers that utilized the studied VANET simulators in their research. As shown, the currently two most popular VANET simulators are GrooveNet and NCTUns.

## 6. Conclusion

The increasing popularity and attention in VANETs has prompted researchers to develop accurate and realistic simulation tools. In this paper, we make a survey of several publicly available mobility generators, network simulators, and VANET simulators.

The mobility generators studied include SUMO, MOVE, CityMob, FreeSim, STRAW, Netstream, and VanetMobiSim. SUMO, MOVE, STRAW, and VanetMobiSim all have good software features and traffic model support. However, only VanetMobiSim provides excellent trace support. CityMob is good in software features and traffic model support. FreeSim exhibits good software characteristics but is limited in other features.

Among the network simulators studied, ns-2, GloMoSim, JiST/SWANS, and SNS all exhibit good software support. However, both ns-2 and GloMoSim are poor in scalability while JiST/SWANS is harder to use than others. In fact, all network simulators do not specifically address VANET scenarios and requirements, such as the consideration of 802.11p, obstacles, vehicular traffic flow, etc.

Finally, in terms of VANET simulators, we studied TraNS, GrooveNet, NCTUns, and MobiREAL. TraNS and MobiREAL both involve the coupling of a VANET mobility generator with a network simulator. GrooveNet and NCTUns, however, are self-contained simulators with GrooveNet capable of supporting hybrid simulations, i.e., communications between simulated vehicles and real vehicles. A survey of recently published papers shows that GrooveNet and NCTUns are more frequently used for VANET simulations than others. Although these four VANET simulators are now publicly available, we realize that further refinement, extensions, and contributions are needed before they can be widely accepted and used for supporting VANET research.

## Acknowledgments

This work was partially supported by the Ministerio de Educación y Ciencia, Spain, under Grant TIN2008-06441-C02-01, and by the Fundación Antonio Gargallo, under Grant 2008/B010.

## References

1. Bechler M, Franz WJ, Wolf L. Mobile Internet access in FleetNet. In *Verteilten Systemen KiVS 2003*, 2003.
2. Toh C-K. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall: Upper Saddle River, NJ, USA, 2001.
3. Cavin D, Sasson Y, Schiper A. On the accuracy of MANET simulators. In *Proceedings of the 2nd ACM International Workshop on Principles of Mobile Computing*. ACM: New York, NY, USA, 2002; 38–43.
4. Yoon J, Liu M, Noble B. Random waypoint considered harmful. *Proceedings of IEEE INFOCOMM 2003*, San Francisco, California, USA, 30 March–3 April 2003.
5. McTrans. Traffic Software Integrated System-Corridor Simulation (TSIS-CORSIM). Center for Microcomputers in Transportation (McTrans), 2007. Available at: http://mctrans.ce.ufl.edu/featured/tsis/
6. Quadstone Paramics, 2008. Available at: http://www.paramics-online.com/
7. Capra L, Emmerich W, Mascolo C. Carisma: context-aware reflective middleware system for mobile applications. *IEEE Transactions on Software Engineering* 2003; **29**: 929–945.
8. VISSIM, 2008. Available at: http://www.ptvamerica.com/vissim.html
9. Scalable Network Technologies. Qualnet. Scalable Network Technologies, Inc., 2006. Available at: http://www.scalable-networks.com/products/download.php
10. OPNET Technologies, 2008. Available at: http://www.opnet.com/
11. Krajzewicz D, Rossel C. Simulation of Urban MObility (SUMO). German Aerospace Centre, 2007. Available at: http://sumo.sourceforge.net/index.shtml
12. MOVE (MObility model generator for VEhicular networks): Rapid Generation of Realistic Simulation for VANET, 2007. Available at: http://lens1.csie.ncku.edu.tw/MOVE/index.htm
13. Martinez FJ, Cano JC, Calafate CT, Manzoni P. Citymob: a mobility model pattern generator for VANETs. In *IEEE Vehicular Networks and Applications Workshop (Vehi-Mobi, held with ICC)*, Beijing, China, May 2008.
14. STRAW - STreet RAndom Waypoint - vehiclar mobility model for network simulations (e.g., car networks), 2008. Available at: http://www.aqualab.cs.northwestern.edu/projects/STRAW/index.php
15. FreeSim, 2008. Available at: http://www.freewaysimulator.com/
16. Mori H, Kitaoka H, Teramoto E. Traffic simulation for predicting traffic situations at expo 2005. *R&D Review of Toyota CRDL* 2006; **41**(4): 45–51.
17. Haerri J, Fiore M, Fethi F, Bonnet C. VanetMobiSim: generating realistic mobility patterns for VANETs. Institut Eurécom and Politecnico Di Torino, 2006. Available at: http://vanet.eurecom.fr/
18. Fall K, Varadhan K. ns notes and documents. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 2000. Available at: http://www.isi.edu/nsnam/ns/ns-documentation.html
19. Martin J. GloMoSim. Global mobile information systems simulation library. UCLA Parallel Computing Laboratory, 2001. Available at: http://pcl.cs.ucla.edu/projects/glomosim/
20. Walsh K, Sirer EG. A staged network simulator (SNS). Computer Science Department, Cornell University, 2003. Available at: http://www.cs.cornell.edu/people/egs/sns/
21. JiST/SWANS: Java in Simulation Time/Scalable Wireless Ad hoc Network Simulator, 2004. Available at: http://jist.ece.cornell.edu/
22. The Georgia Tech Network Simulator (GTNetS), 2008. Available at: http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/
23. Piorkowski M, Raya M, Lugo AL, Papadimitratos P, Grossglauser M, Hubaux J-P. TraNS (Traffic and Network Simulation Environment). Ecole Polytechnique Fédérale de Lausanne, EPFL, Switzerland, 2007. Available at: http://trans.epfl.ch/
24. NCTUns 5.0, 2008. Available at: http://nsl10.csie.nctu.edu.tw/
25. Mangharam R, Weller D, Rajkumar R, Mudalige P, Bai F. GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks. Carnegie Mellon University, 2006. Available at: http://www.seas.upenn.edu/rahulm/Research/GrooveNet/
26. MobiREAL, 2008. Available at: http://www.mobireal.net/
27. Choffnes DR, Bustamante FE. Modelling vehicular traffic and mobility for vehicular wireless networks. *Technichal Report*, Department of Computer Science, Northwestern University, NWU-CS-05-03, July 2005.
28. Salem HH, Chrisoulakis J, Papageorgiou M, Elloumi N, Papadakos P. The use of METACOR tool for integrated urban and interurban traffic control. evaluation in corridor peripherique, Paris. In *Proceedings of Vehicle Navigation and Information Systems Conference*, 1994; 645–650.
29. Taylor NB. The CONTRAM dynamic traffic assignment model. *Networks and Spatial Economics* 2003; **3**: 297–322.
30. Nagel K, Schleicher A. Microscopic traffic modeling on parallel high performance computers. *Parallel Computing* 1994; **20**: 125–146.
31. Krauss S. Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics. *Ph.D. Dissertation*, 1998.
32. Treiber M, Hennecke A, Helbing D. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E* 2000; **62**: 1805.
33. Fiore M, Haerri J, Filali F, Bonnet C. Vehicular mobility simulation for VANETS. In *Proceedings of the 40th Annual Simulation Symposium (ANSS 2007)*, Norfolk, Virginia, March 2007.
34. CANU Research Group. CanuMobiSim (Mobility Simulation Environment). Institute of Parallel and Distributed Systems (IPVS). University of Stuttgart, 2001. Available at: http://canu.informatik.uni-stuttgart.de/mobisim/index.html
35. VISUM, 2008. Available at: http://www.ptvamerica.com/visum.html
36. ArcView, 2008. Available at: http://esri.com/software/arcview/
37. OpenStreetMap: The Free Wiki World Map, 2008. Available at: http://www.openstreetmap.org/
38. Car-to-car cooperation for vehicular ad-hoc networks. An AquaLab Project, 2002. Available at: http://www.aqualab.cs.northwestern.edu/projects/C3.html.
39. Real mobility traces for ns2 and Jist/SWANS simulations, 2008. Available at: http://lcawww.epfl.ch/piorkowski/real_mobility_traces.html
40. The CMU Monarch Project. The CMU Monarch Project's wireless and mobility enhancements to ns. Carnegie-Mellon University, 2001. Available at: http://www.monarch.cs.cmu.edu
41. Chen Q, Schmidt-Eisenlohr F, Jiang D, Torrent-Moreno M, Delgrossi L, Hartenstein H. Overhaul of ieee 802.11 modeling and simulation in ns-2. In *MSWiM'07: Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*. ACM: New York, NY, USA, 2007; 159–168.
42. UCLA Parallel Computing Laboratory. Parsec: Parallel Simulation Environment for Complex Systems, 2008. Available at: http://pcl.cs.ucla.edu/projects/parsec/
43. SWANS User Guide, 2004. Available at: http://jist.ece.cornell.edu/docs/040319-swans-user.pdf
44. Walsh K, Sirer EG. Staged simulation: A general technique for improving simulation scale and performance. *ACM Transactions on Modeling and Computer Simulation* 2004; **14**(2): 170–195.
45. Sichitiu M, Kihl M. Inter-vehicle communication systems: a survey. *IEEE Communications Surveys and Tutorials* 2008; **10**(2): 88–105.

DOI: 10.1002/wcm

46. Wegener A, Piórkowski M, Raya M, Hellbrück H, Fischer S, Hubaux J-P. Traci: an interface for coupling road traffic and network simulators. In *CNS'08: Proceedings of the 11th Communications and Networking Simulation Symposium*. ACM: New York, NY, USA, 2008; 155–163.
47. Roadnav, 2004. Available at: http://roadnav.sourceforge.net/
48. Qt Cross-Platform Application Framework, 2008. Available at: http://trolltech.com/products/
49. IEEE xplore, 2008. Available at: http://ieeexplore.ieee.org/Xplore/dynhome.jsp
50. google scholar, 2008. Available at http://scholar.google.com/

## Authors' Biographies

**Francisco J. Martinez** is an associate professor in the Department of Computers and Systems Engineering at the University of Zaragoza in Spain. He graduated in Computer Science and Documentation at the Technical University of Valencia in 1996 and 1999, respectively. His current research interests include VANET simulation, intelligent transportation systems, traffic security, vehicle-to-vehicle (V2V), and vehicle-to-infrastructure (V2I) communications.

**Chai Keong Toh** received his Ph.D. in Computer Science from Cambridge University (1996) and earlier E.E. degrees from Manchester University (1991) and Singapore Polytechnic (1986). He was a visiting professor at YALE (2007), NTU (Taiwan), Fudan (China), KTH (Sweden), KNU (Korea), and Osaka University (Japan). Since 2004, he has been an Honorary Full Professor at the University of Hong Kong. Previously, he was a tenured Chair Professor at the University of London and the Director of Research (Communication Systems) at TRW Tactical Systems Inc., USA. Earlier on, he led the DARPA Deployable *Ad Hoc* Networks program at Hughes Research. Dr Toh had authored two books: (a) *Ad Hoc Mobile Wireless Networks* (Prentice Hall, ISBN 0130078174, 2001), and (b) *Wireless ATM* (Kluwer, ISBN 079239822X, 1997). He has consulted for various companies, including KDDI (Japan), Hughes (USA), and Mitsubishi (USA). Dr Toh is a Fellow of the British Computer Society (BCS), New Zealand Computer Society (NZCS), Hong Kong Institution of Engineers (HKIE), and Institution of Electrical Engineers (IEE). He is a recipient of the 2005 IEEE Institution Kiyo Tomiyasu Technical Field Award (for pioneering contributions to communication protocols in *ad hoc* mobile networks). In 2009, he was named an IEEE Fellow.

**Juan-Carlos Cano** is an associate professor in the Department of Computer Engineering at the Polytechnic University of Valencia (UPV) in Spain. He earned an M.Sc. and a Ph.D. in computer science from the UPV in 1994 and 2002 respectively. Between 1995 and 1997 he worked as a programming analyst at IBM's manufacturing division in Valencia. His current research interests include power aware routing protocols and quality of service for mobile *ad hoc* networks and pervasive computing. You can contact him at jucano@disca.upv.es.

**Carlos T. Calafate** graduated with honors in Electrical and Computer Engineering at the University of Oporto (Portugal) in 2001. He received his Ph.D. in Computer Engineering from the Technical University of Valencia in 2006, where he has worked as an assistant professor since 2005. He is a member of the Computer Networks research group (GRC). His research interests include mobile and pervasive computing, security and QoS on wireless networks, as well as video coding and streaming.

**Pietro Manzoni** received the M.S. degree in computer science from the 'Universitá degli Studi' of Milan, Italy, in 1989, and the Ph.D. in computer science from the Polytechnic University of Milan, Italy, in 1995. He is an associate professor of computer science at the Polytechnic University of Valencia, Spain. His research activity is related to wireless networks protocol design, modeling, and implementation. He is member of the IEEE.

DOI: 10.1002/wcm