A framework to simulate VANET scenarios with SUMO

Florent Kaisser, Christophe Gransart, Mohamed Kassab and Marion Berbineau
Université Lille Nord de France
IFSTTAR-LEOST
F-59650 Villeneuve d'Ascq, France
firstname.lastname@ifsttar.fr

Abstract

Vehicular Ad hoc Networks (VANET) are a special kind of Mobile Ad-Hoc Networks (MANET) adapted to the communications between vehicles. Several specific protocols to VANETs have been developed to improve performances and satisfy vehicular application needs.
To evaluate a protocol for VANET, some realistic mobility models are needed. Unfortunately, such models are not provided by OPNET Modeler. In this work, we propose a framework that enhances OPNET simulation scenario using realistic vehicular mobility models. This framework makes use of the open source software called "Simulation of Urban MObility" (SUMO) and the "input trajectory files" feature of OPNET.

Introduction

Starting with the idea of making driving safer by inter-vehicle communication, the concept of vehicular networks or vehicular ad hoc networks (VANETs) has been extended to a large collection of various applications that can profit from wireless communication between vehicles. Vehicles are not only communicating between each other, but are also getting information from and sending data to infrastructure units.

This particular context has several specificities like the deployment of safety applications that have critical requirements regarding communications. Classical ad hoc routing protocol (AODV, OLSR, GPSR) are usable in VANET, but they are not adapted to the specificities of these networks. Several specific protocols to VANETs have been developed to improve performances and satisfy vehicular application needs. To evaluate scalability and efficiency of a protocol for VANET, some simulations with realistic mobility model are needed. Numerous works use various tools to evaluates their proposals. In this paper we propose to use OPNET to simulate VANET by associating it with a vehicular mobility simulator.

In a first section, we propose an overview of VANET simulation by describing current simulation tools. Afterwards, in a second section, we introduce our framework to simulate VANET with OPNET Modeler. Finally, we use our work to evaluate a simple dissemination protocol for VANET by simulation of adapted scenarios.

VANET simulation Overview

In this section, we give a brief overview of vehicular ad hoc network and tools to simulate them. Like other kind of networks, simulating VANET is necessary to validate protocols and applications. In addition, the total number of vehicles in a network can be large, so evaluating the scalability is central.

Many tools allow to simulate VANET. In many case, this tool is a vehicular mobility model associate with a network simulator. In some case, vehicular mobility model is a simple random way point mobility model, but it restricts the movement of vehicles to real roads. To improve realism of the simulation, a *car-following* model can be added, allowing to compute the vehicle acceleration (then speed and position) according to neighbor vehicles in the same lane or adjacent lanes.
Next, we describe some network simulators associated to a vehicular mobility model. A more exhaustive list of tools for VANET is depicted in a dedicated book [1].

SWANS++ [2] is the network simulator SWANS (from JiST/SWANS [3] project) extended with a vehicular mobility model, STRAW [4] (Street Random Way point). STRAW uses a simple random way point mobility model but it takes in account a real street map loaded from TIGER [5] (Topologically Integrated Geographic Encoding and Referencing, an US street database) data files.

GrooveNet [6] (originally known as GrooveSim) is a network and mobility simulator for real and simulated vehicles. GrooveNet, originally as an extension of roadnav [7] open-source simulator, can load real street map from TIGER database and includes a car-following model.

TraNS [8] (Traffic and Network Simulation Environment) was the first work to combine the network simulator ns-2 [9] with a vehicular mobility simulator SUMO. We introduce SUMO below. The link between the two simulators is done with a parser that reads the output of SUMO and converts it to a suitable format for ns-2. Other works like MOVE [10] GUI and TRACEEXPORTER tool (module of SUMO) allow to link ns-2 with SUMO.

In Veins [11] (Vehicles in Network Simulation), SUMO is paired with an other network simulator OMNET++ [12]. It creates a bidirectional communication between SUMO and OMNET++. Thus, the network simulator can react to events from the mobility simulator.

iTETRIS simulation platform [13] use ns-3 [14] as network simulator and SUMO. The two open source simulation platforms are connected with a central module called iTETRIS Control System (iCS). This approach permits to have the best performance in terms of scalability and modularity.

OPNET Modeler enables both *random mobility* and *trajectory mobility*. The last-mentioned is setting with one trajectory file per node. This method allows to elaborate complex node movements. Unfortunately, interaction between nodes are not taken in account, so realistic mobility is limited.

In this paper we propose a framework to simulate complex mobility, in our case moving vehicles on a road, using an extern microscopic traffic simulation called SUMO.

A framework to simulate VANET

A microscopic traffic simulator computes the position of all vehicles on road at some time *t* from vehicles and road characteristics.
These positions are used to show vehicles on a graphical user interface (GUI) or writen in a file (*dump* file).

Input data can be initials positions, vehicle characteristics (acceleration, maximum speed, etc), road network,way to go,

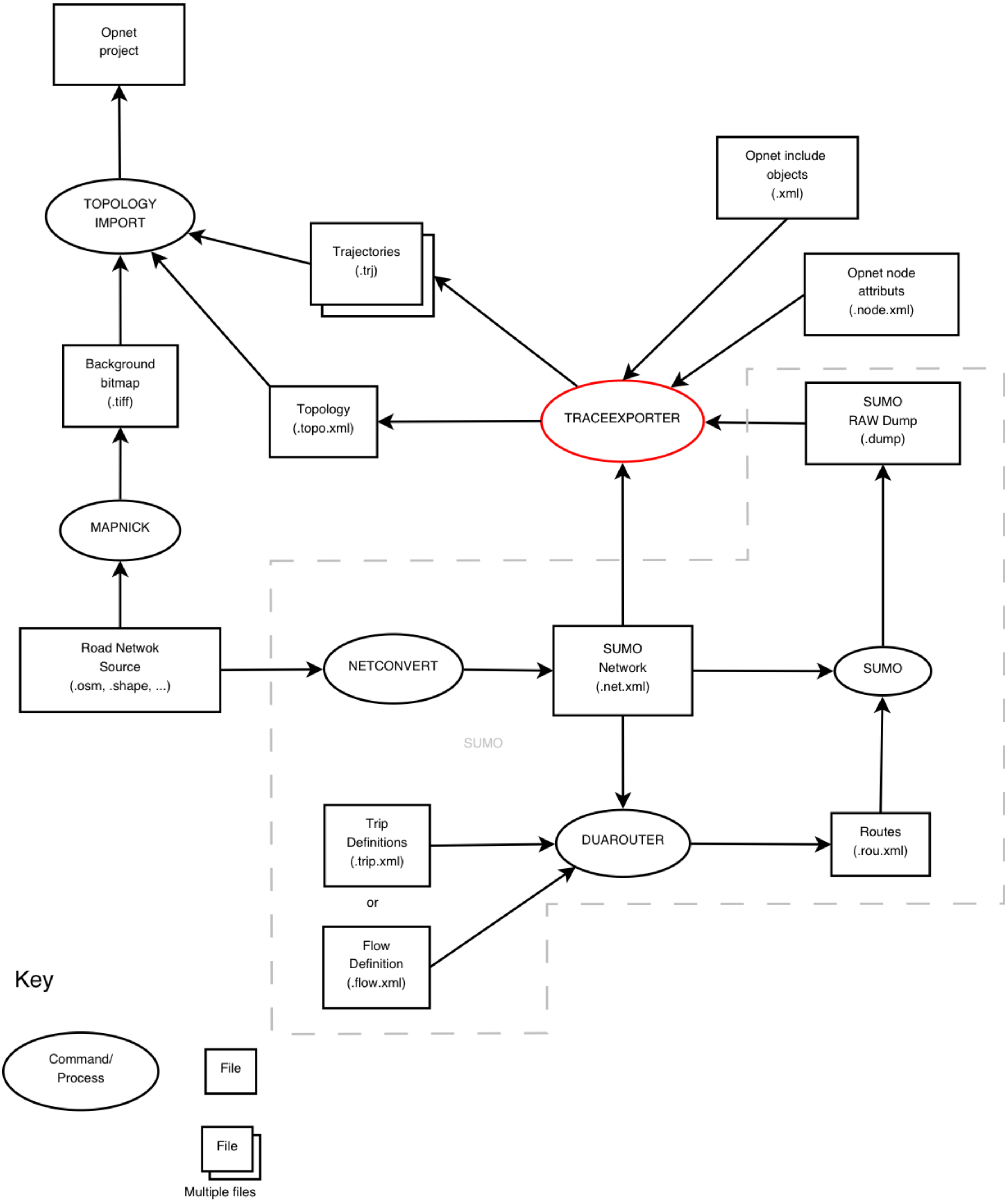


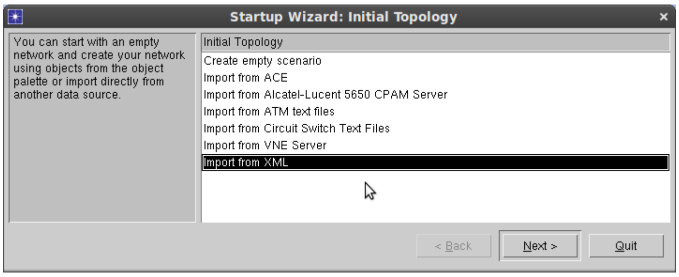
Figure 1: Process to generate vehicular mobility with SUMO

position of destination, speed limits, etc. SUMO [15] is a free implementation of such a simulator and supports car-following model (issue of [16][17][18][19]). SUMO implementation consists of several modules (or commands), with the SUMO module in the center (Figure 1). Input data of SUMO is one or several road network files (*network* file) and one route file. The road network file can be generated from various input formats with NETCONVERT module. These input formats can be osm files, from Open Street Map [20] database, or *shape* files, from TIGER [5] database. Many other formats are available.

A route file describes both the characteristics of vehicles on the road and the path taken by the cars (edge sets of network road). Like net files, road files can be generated with DUAROUTER module from two kinds of input: *trip* or *flow* definition. Each trip consists at least of the starting, the ending edge and the departure time. Then DUAROUTER computes the shortest path between theses edges. Each trip is associated with one vehicle. Flow is mostly the same approach as using trip definitions, but one may join vehicles having the same departure and arrival edge using this method. Moreover, initial position of all vehicles are computed randomly (from a simple discrete uniform distribution).

Generate simple scenario

From network and route files, sumo module simulates all the positions at several times (e.g. every seconds). These positions are written in a *dump* file. This file is used to generate file(s) in a format understood by network simulator softwares. In our case, we need a converter to generate files for OPNET Modeler. This is achieved by developing our extension TRACEEXPORTER[1]. In addition of dump file, two other input XML files are added: *node attributes* and *include objects*. The scheme of these XML files are the same as the files generated by OPNET exporter feature. The node attribute file contains the attributes of each vehicle node (model name, parameters, icon, etc). The include object file contains the description of all objects to be added to the scenario, e.g. a backbone or statics equipments.


Startup Wizard: Initial Topology

You can start with an empty network and create your network using objects from the object palette or import directly from another data source.

Initial Topology
Create empty scenario
Import from ACE
Import from Alcatel-Lucent 5650 CPAM Server
Import from ATM text files
Import from Circuit Switch Text Files
Import from VNE Server
Import from XML

< Back | Next > | Quit

Figure 2: Topology import with new scenario assistant

TRACEEXPORTER generates one *trajectories* Opnet file for each vehicle and one *topology* OPNET file. The trajectories file is the same file generated by the GUI of Modeler when user defines a trajectorie for a node. Topology file contains the scenario description with nodes description (name and initial vehicles position from SUMO dump file, and attributes from node attribute file) and other objects description (from include object file). In addition of this file, a *background* bitmap can be generated from network road file with adequate tool (e.g. MAPNICK for *osm* file).

Once trajectories files are copied to a OPNET model directory, the user calls the Topology importer feature from OPNET Modeler (Topology menu, Topology import, from XML files...), or creates a new scenario with XML import option (Figure 2). User should obtain a results like shown in Figure 3. Discrete Event Simulation can be run from this point.

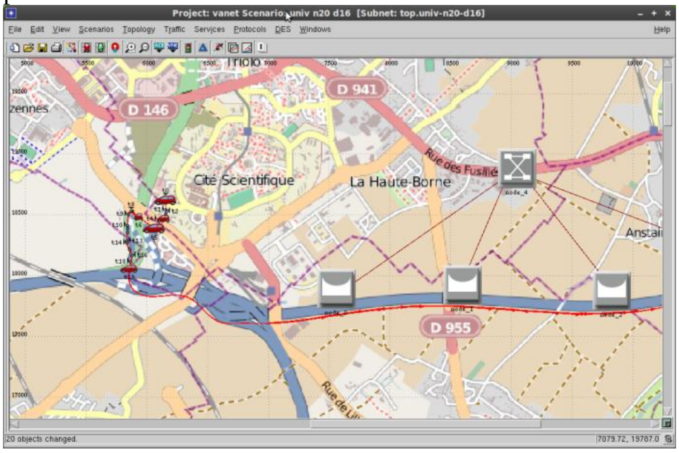

Project: vanet Scenario_univ n20 d16 [Subnet: top.univ-n20-d16]
File Edit View Scenarios Topology Traffic Services Protocols DES Windows Help
D 146
D 941
Cité Scientifique
La Haute-Borne
Rue des Fusillés
D 955
20 objects changed.

Figure 3: VANET in Opnet Modeler with our framework

Generate multiple scenarios

We added a feature to TRACEEXPORTER allowing to generate multiple scenarios at the same time. A *scenario file* defines all scenarios to generate. Each scenario contain links to one input file for the road network (road network file), one for the route definitions (route file, trip file or flow file), one for the include objects and one for the node attributes (Figure 4). The different scenario can use same or different input files. For examples, to generate scenarios with different number of vehicles (50, 100, 150), three scenarios use the same network and node attributes files, but three different flow files containing three number of vehicles to generate. Moreover, TRACEEXPORTER runs automatically the commands like SUMO and DUAROUTER (for this reson, SUMO does not appear on Figure 4). After execution of TRACEEXPORTER, one topology file is generated for each scenario in different directories with their trajectories files associated.

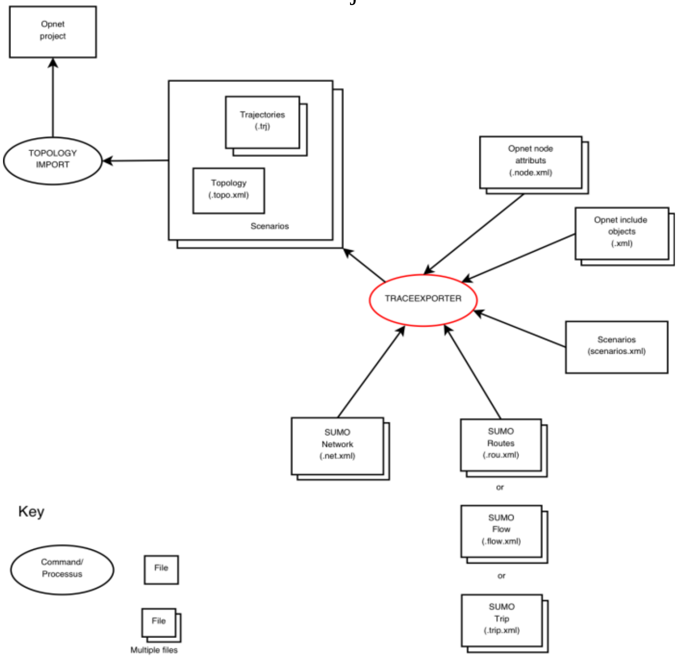


Figure 4: Multiple scenarios generation with traceexporter

[1] Extension *tranceexporter* already exists, but it only for NS-2. We have preferred re-develop a similar extension for both NS-2 and Opnet simulators. Moreover our implementation is easier to extend.

Dissemination of informations in VANET

Several applications for VANET need to disseminate information on the network. For security, an alert is spread to neighbor vehicles or the current informations of vehicles (like the position, speed or other information from vehicle probes) can be sent to all neighbors to increase driver visibility.

We take this application as an instance of a VANET simulation. Therefor, we add a process model for OPNET Modeler to simulate a dissemination of information in VANET. This process is above *wireless_lan_mac*. A new packet format is defined, containing: vehicle ID, type of information, position, number of hop, lifetime of the information, sequence number. This packet is the payload of a MAC packet sent by MAC layer.

The process sends a packet every 500 ms. We define a simple forwarding scheme (flooding): when the process receives a packet not received in the past, the packet is re-sent to the neighbors. Thus, all nodes in the network receive informations contained in the packet.

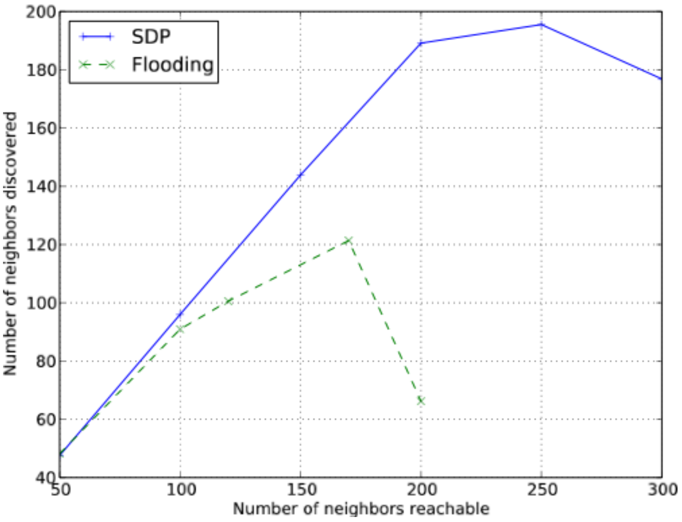


Figure 5: Neighboor discovery with simple dissemination protocols

With a high density of vehicles, the number of packet sent in the network can reach the network capacity, packets are no longer forwarded, then some vehicles do no receive information. To avoid this, we define a forwarding scheme to limit the number of packet sent. Several existing works [21][22][23][24][25] propose an achievement of dissemination in VANET. In this paper, we simulate (in addition to flooding) a simple distance-based protocol (SDP).

On message reception, a vehicle starts a timer T inversely proportional to the distance from previous sender :

$$T = t_{max} \cdot \left(1 - \frac{d}{R}\right)$$

with t_{max} the maximum contention time, d the distance from previous sender, R the radio range. When the timer reaching zero the node forward the packet. Other waiting nodes, receive this packet and cancel their timer. Thus, we reduce the number of packet sent in the network.

We simulate flooding and SDP protocol with several number of vehicles. In Figure 5 we see a good improvement with SDP protocol: the maximum number of discovered neighbor with flooding is about 120, whereas the maximum is close to 200 with SDP.

Conclusion

In this paper we have proposed a framework to simulate VANET from OPNET Modeler and SUMO. Our proposal allows to use a realistic mobility model to compute movement of vehicles on a real road with OPNET Modeler to effectively simulate communication networks and with many existing models. The aim of this framework is to be modular and easy to use by exploiting judicious feature of Modeler like topology import.

But simulation of VANET with several number of vehicles or random seed require many OPNET scenarios and an equivalent number of topology import, which limits a good protocols evaluation. In future works, we wish to integrate in OPNET modeler a direct and bidirectional communication with SUMO allowing us to define more complex simulation scenarios.

Source code of this work can be downloaded at https://bitbucket.org/florent_k/traceexporter

Bibliographie

[1] Olariu, S. and Weigle, M.A.C., Vehicular Networks: From Theory to Practice, 2009
[2] SWANS++ : Extensions to the Scalable Wireless Ad-hoc Network Simulator, http://www.aqualab.cs.northwestern.edu/projects/swans++/
[3] Jist / Swans : Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator, http://jist.ece.cornell.edu
[4] Choffnes, D.R., Straw-an integrated mobility and traffic model for vanets, 2005
[5] TIGER : Topologically Integrated GEographic Encoding and Referencing, http://www.census.gov/geo/www/tiger/
[6] Rahul Mangharam and Daniel S. Weller and Daniel D. Stancil and Ragunathan Rajkumar and Jayendra S. Parikh, GrooveSim: A Topography-Accurate Simulator for Geographic Routing in Vehicular Networks, 2005
[7] Roadnav, http://roadnav.sourceforge.net
[8] Piorkowski, M. and Raya, M. and Lugo, A.L. and Papadimitratos, P. and Grossglauser, M. and Hubaux, J.P., TraNS: realistic joint traffic and network simulator for VANETs, 2008
[9] NS2 : the network simulator, http://www.isi.edu/nsnam/ns
[10] Rapid generation of realistic mobility models for VANET,
[11] Sommer, C. and Yao, Z. and German, R. and Dressler, F., Simulating the influence of IVC on road traffic using bidirectionally coupled simulators, 2008
[12] OMNeT++ Community Site, http://www.omnetpp.org
[13] iTetris Test-beds, http://www.ict-itetris.eu/platform.htm
[14] The ns-3 network simulator, http://www.nsnam.org
[15] SUMO - Simulation of Urban MObility, http://sumo.sourceforge.net
[16] Krauss, S., Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics, 1998
[17] Wagner, P. and Lubashevsky, I., Empirical basis for car-following theory development, 2003
[18] Kerner, B.S., Introduction to modern traffic flow theory and control: the long road to three-phase traffic theory, 2009

[19] Treiber, Martin and Hennecke, Ansgar and Helbing, Dirk , Congested traffic states in empirical observations and microscopic simulations, 2000
[20] Haklay, M.M. and Weber, P., OpenStreetMap: User-generated street maps, 2008
[21] Kim, T.H. and Hong, W.K. and Kim, H.C., An effective multi-hop broadcast in vehicular ad-hoc network, 2007
[22] Briesemeister, L. and Schafers, L. and Hommel, G., Disseminating messages among highly mobile hosts based on inter-vehicle communication, 2000
[23] Hrizi, F. and Filali, F., On congestion-aware broadcasting in V2X networks, 2009
[24] Min-te Sun, W.F. and Lai, T.H. and Yamada, K. and Okada, H. and Fujimura, K., GPS-based message broadcasting for inter-vehicle communication, 2000
[25] Hrizi, F. and Filali, F., Achieving broadcasting efficiency in V2X networks with a distance-based protocol, 2009