



HAL
open science

Simulations of VANET Scenarios with OPNET and SUMO

Florent Kaiser, Christophe Gransart, Marion Berbineau

► **To cite this version:**

Florent Kaiser, Christophe Gransart, Marion Berbineau. Simulations of VANET Scenarios with OPNET and SUMO. Nets4Cars/Nets4Trains, Apr 2012, France. 9p. hal-00985936

HAL Id: hal-00985936

<https://hal.science/hal-00985936>

Submitted on 30 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simulations of VANET scenarios with OPNET and SUMO

Florent Kaiser, Christophe Gransart, and Marion Berbineau

Univ Lille Nord de France, F-59000, Lille,
IFSTTAR, LEOST, F-59650, Villeneuve d'Ascq
Email: florent.kaiser@ifsttar.fr, christophe.gransart@ifsttar.fr,
marion.berbineau@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 Modeler 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 Modeler.

Keywords: VANET, Opnet, SUMO, simulation, framework, dissemination.

1 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 evaluate their proposals. In this paper we propose to use Opnet Modeler 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.

2 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 to simulate VANET is depicted in a dedicated book [?] and a paper [?].

SWANS++ [?] is the network simulator SWANS (from JiST/SWANS [?] project) extended with a vehicular mobility model, STRAW [?] (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 [?] (Topologically Integrated Geographic Encoding and Referencing, an US street database) data files.

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

TraNS [?] (Traffic and Network Simulation Environment) was the first work to combine the network simulator ns-2 [?] 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 [?] GUI and *traceexporter* tool (module of SUMO) allow to link ns-2 with SUMO.

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

iTETRIS simulation platform [?] use ns-3 [?] 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. To the best of our

knowledge, no tool for Opnet Modeler exists allowing to simulate VANET with a realistic mobility model.

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

3 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 written in a file (*dump file*). These simulators use a realistic model to describe the vehicles moving. This model computes the ideal acceleration, velocity and position for each vehicle any time, therefore avoid the physical collisions between vehicle.

Input data can be initials positions, vehicle characteristics (acceleration, maximum speed, etc), road network, way to go, position of destination, speed limits, etc. SUMO [?] is a free implementation of such a simulator and supports car-following model (issue of [?, ?, ?, ?]). SUMO implementation consists of several modules (or commands), with the SUMO module in the center (Figure ??). Input data of SUMO is one or several *road network files* 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 [?] database, or *shape files*, from TIGER [?] 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 road network files, route files can be generated with a other module called *duarouter* 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).

3.1 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. This extension implement the data flow diagram shown in Figure ??.

¹ Extension *traceexporter* already exists, but it only for ns-2. We have preferred redevelop a similar extension for both ns-2 and Opnet Modeler simulators. Moreover our implementation is more modular and therefore more easier to extend.

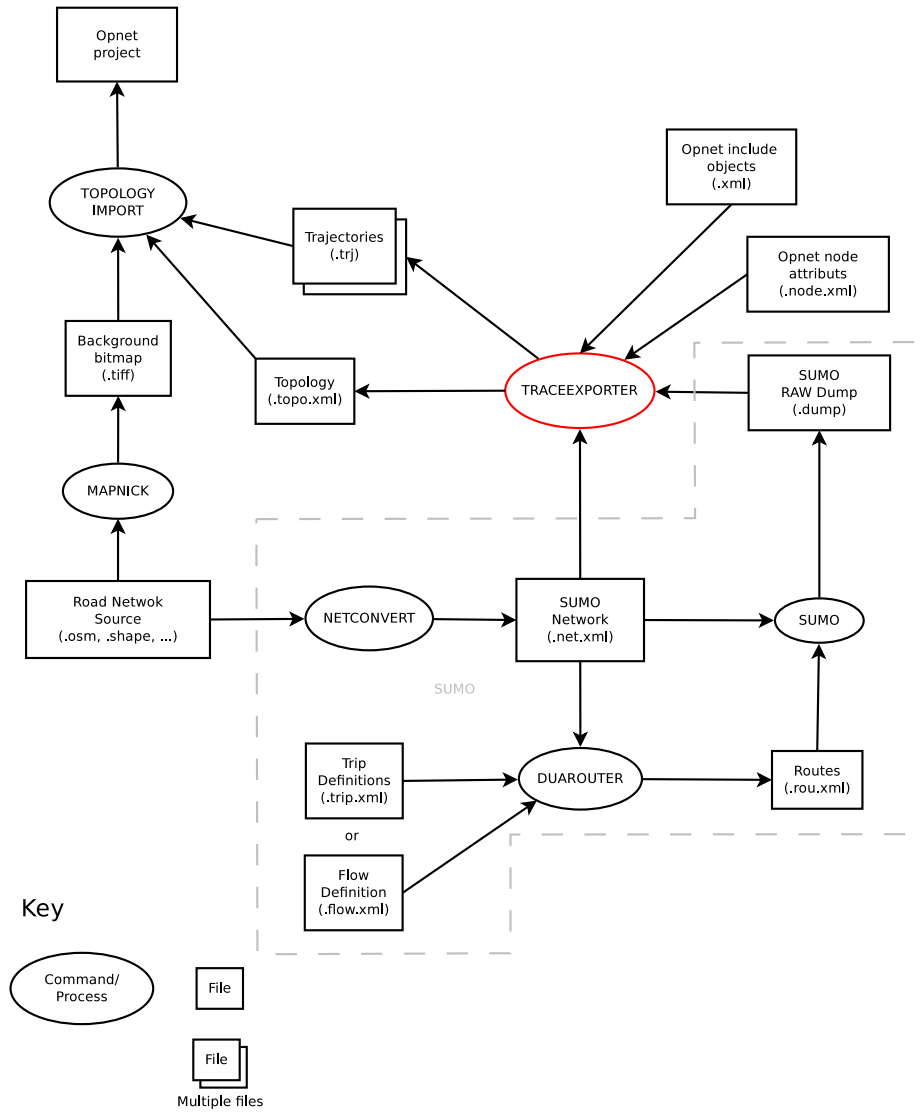


Fig. 1. Process to generate vehicular mobility with SUMO

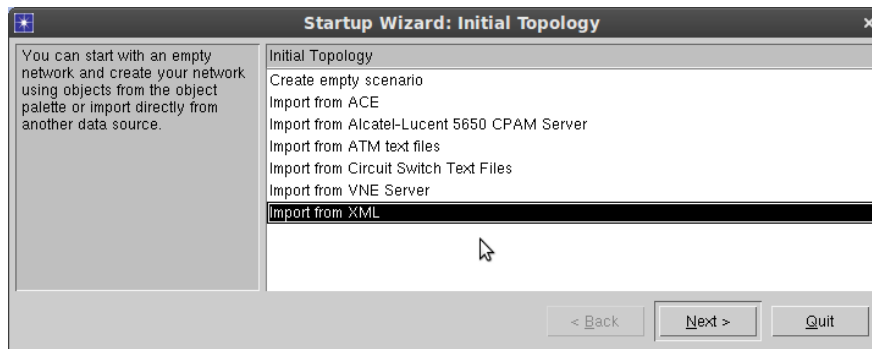


Fig. 2. Topology import with new scenario assistant

input XML files are added: node attributes and include objects. The scheme of these XML files are the same as the files generated by Modeler 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 static equipments.

traceexporter generates one *trajectories file* for each vehicle and one *topology file*. The trajectories file is the same file generated by the GUI of Modeler when user defines a trajectory 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 ??). User should obtain a results like shown in Figure ?. Discrete Event Simulation can be run from this point.

3.2 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 ??). 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 reason, SUMO does not appear on Figure ??).

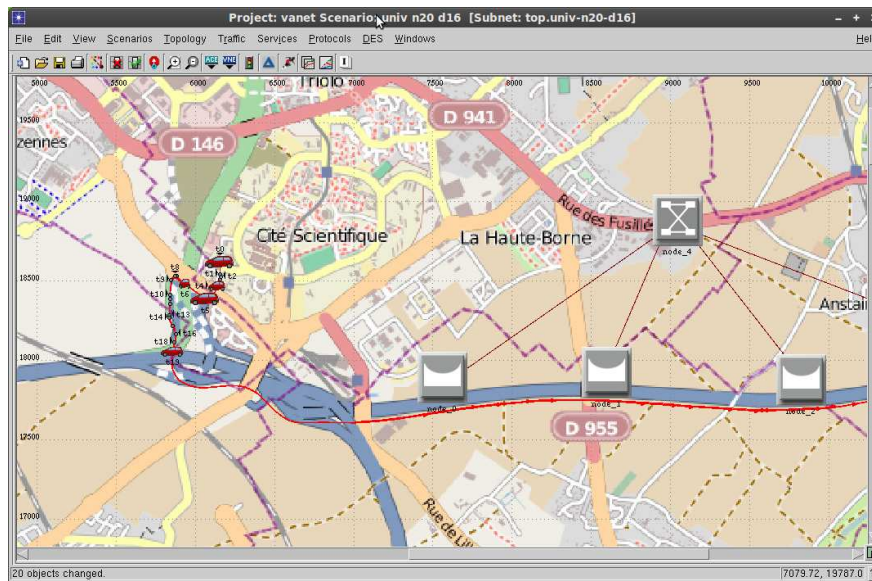


Fig. 3. VANET in Opnet Modeler with our framework

After execution of *traceexporter*, one topology file is generated for each scenario in different directories with their trajectories files associated.

4 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. Therefore, 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.

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 not receive information. To avoid this, we define a forwarding scheme to limit the number of packet sent. Several existing works [?, ?, ?, ?] propose an

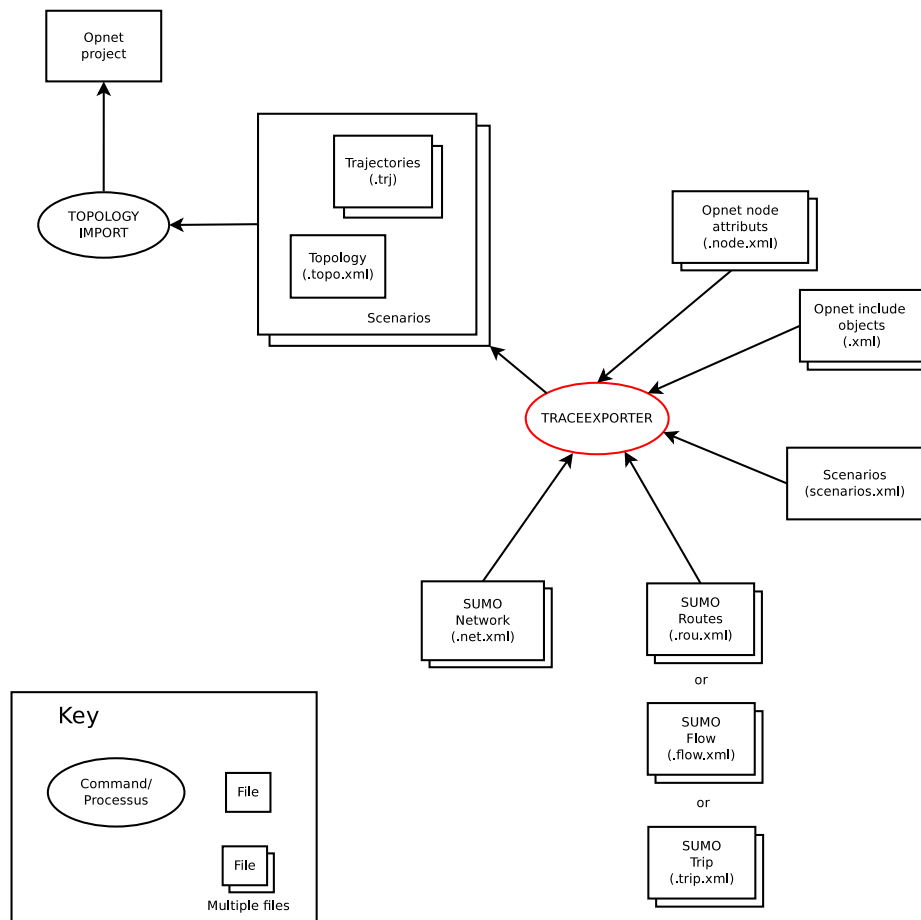


Fig. 4. Multiple scenarios generation with *traceexporter*

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 :

$$T = t_{max} * (1 - \frac{d}{R})$$

with t_{max} the maximum waiting time, d the distance from previous sender, R the radio range. In the simulation, $R = 550 m$ and $t_{max} = 50 ms$. 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 ?? 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.

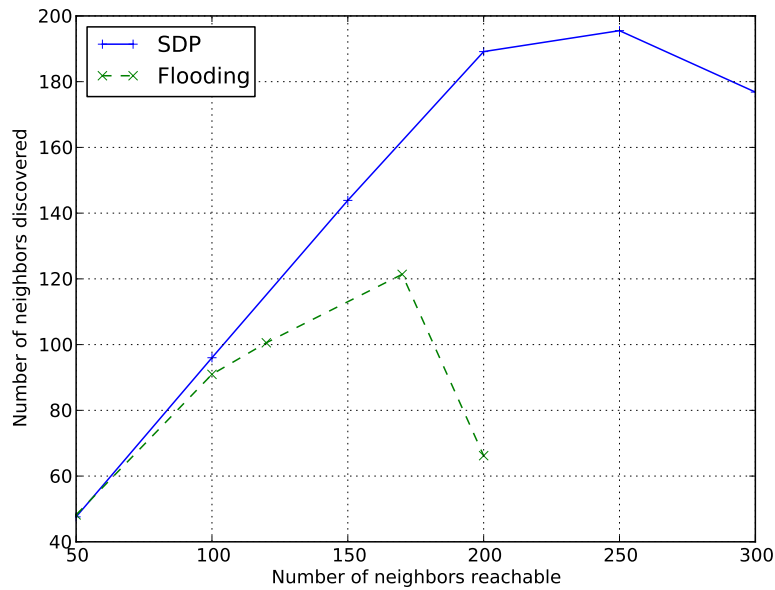


Fig. 5. Neighbor discovery with simple dissemination protocols

5 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 Modeler 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.