



HAL
open science

HINTS: A novel approach for realistic simulations of vehicular communications

Sofiane Zemouri, Sara Mehar, Sidi-Mohammed Senouci

► **To cite this version:**

Sofiane Zemouri, Sara Mehar, Sidi-Mohammed Senouci. HINTS: A novel approach for realistic simulations of vehicular communications. 2012 Global Information Infrastructure and Networking Symposium (GIIS), Dec 2012, Choroni, Venezuela. pp.1-6, 10.1109/GIIS.2012.6466764 . hal-02557965

HAL Id: hal-02557965

<https://hal.science/hal-02557965>

Submitted on 27 Mar 2022

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.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

HINTS: A Novel Approach for Realistic Simulations of Vehicular Communications

Sofiane Zemouri, Sara Mehar, and Sidi-Mohammed Senouci

DRIVE Labs, University of Burgundy, Nevers, France

Nevers, France

Contact: sofiane.zemouri@gmail.com, {Sara.Mehar, Sidi-Mohammed.Senouci}@u-bourgogne.fr

Abstract— One of the main challenges in the Vehicular Ad-Hoc Networks (VANETs) research domain is the simulation of vehicular communications using realistic mobility models. Several efforts have been put lately in this purpose; yet, the proposed models are either inappropriate, or carry significant disadvantages. In this paper, we propose a novel approach for realistic simulations in vehicular networks, inspired from the hierarchical video and image compression technique. We developed HINTS (Hybrid Integration of Network and Traffic Simulators), the correspondent platform, engendered from the integration of SUMO, a traffic simulator and NS-3, a network simulator after adding some supplemental modules to both simulators. Its efficiency and performance are demonstrated in a proof-of-concept study, based on a realistic traffic management scenario.

Keywords- Network simulator, Traffic simulator, Mobility model, intgration, NS-3, SUMO.

I. INTRODUCTION

Vehicular Ad hoc Networks (VANETs) are a special kind of Mobile Ad-Hoc Networks (MANETs) adapted to the communications between vehicles. The great potential of this technology has been acknowledged with the establishment of ambitious research programs and standardization activities on vehicular communication systems worldwide, such as ISO TC204, IEEE (802.11p and 1609.x) and SAE DSRC in the USA, ETSI TC ITS and CEN WG278 in Europe and ARIB T-75 in Japan. In addition, significant evolution inside the vehicles themselves is observed since they are expected to be equipped with high efficiency computing systems and communication capabilities. According to ETSI, in 2027 almost 100% of the vehicles will be equipped with On Board Units (OBUs). OBUs are devices that provide communications among neighboring vehicles (Vehicle to Vehicle) or between vehicles and nearby fixed equipments (Vehicle to Infrastructure). Vehicular networks attract both research and industry communities. Recently, they focus their efforts to grow vehicular communication and networking into maturity by moving it from research field into real implementation (Field Operational Test, FOT); aiming to provide not only more safety in the transportation systems but also other high quality-of-service (QoS) based applications. However, such applications need to be analyzed, tested and

validated using costly real testbeds (FOT), or simply using lower cost simulations.

One simple way of evaluating protocols and applications in VANET is by using communication network simulators. However, this solution appears to be inadequate since existing mobility models, implemented in network simulators, are not realistic. To solve this problem, an approach called offline simulation was introduced; it consists of integrating realistic mobility traces into a network simulator. These traces are extracted from movements of real vehicles that were collected by multiple monitoring systems like induction loops and video surveillance cameras installed in real roads. Another way of extracting these traces is by using a traffic simulator with realistic mobility patterns. The traffic simulator generates the realistic mobility traces prior to the network simulator execution. This approach allows the nodes in the network simulator, to move according to a realistic mobility model that reflects the reality. However, this approach only solves the VANET applications evaluation problem partially. In fact, it is suitable for large scenarios without any unexpected events occurring during simulation time, as it allows the communication between the simulators in one way but not in the opposite way. For example, in a traffic efficiency application which informs the vehicles about the state of the road, a vehicle can decide to change its route to avoid the traffic jam; hence, the need for a bidirectional communication. We can think of a solution where mobility traces are generated after the occurrence of each event, but this will require an enormous amount of time, as the generation of the mobility traces is very slow.

The other major advance in this domain is the online simulation approach. In this case, no static mobility traces are generated by the traffic simulator, but instead, the latter is launched simultaneously with the network simulator. Thus, all the alterations caused by the network simulation can be taken into account in real time. This was made possible with the insertion of an efficient feedback loop which consists in sending the required information (position, speed...) to the network simulator on each simulation step. In the following, we refer to the information sent for each vehicle as a subscription for this vehicle. Although this approach seems to be a good alternative to evaluate and validate Intelligent

Transportation Systems (ITS) applications, it shows some major drawbacks. In fact, it is more suitable for small scenarios on account of the large number of messages exchanged between the simulators on each time step and their important lengths; not to mention the high computational cost and the consequent long simulation times.

As emphasized earlier in this section, the offline simulation approach is suitable for large scenarios without any unexpected events occurring during runtime; where the online simulation is more appropriate for smaller scenarios with a large number of events. In this work, we advocate a novel simulation approach and develop a corresponding optimized platform for realistic simulations of vehicular communications. Our hybrid approach has the advantage of using both offline and online simulations so it overcomes the limitations of one using the other and vice versa.

The remainder of this paper is organized as follows: The next Section is a quick review of the most popular related works in the literature. We describe our approach and its architecture in Sections III and IV respectively; evidence of its performance is given in Section V. Finally, Section VI presents our conclusion and future perspectives.

II. RELATED WORK

The most common classification for the different integration approaches, briefly addressed in the previous section and referred to in [1] for coupling network and traffic simulators, sort the literature solutions into three distinct categories: (i) isolated or *offline* simulators, (ii) *embedded* simulators, and (iii) federated or *online* simulators.

In the first category, the interaction between the network and the traffic simulators is petty or non-existent as the mobility model is bounded to a static mobility trace. MOVE (MObility model generator for VEhicular networks) [2], is a dedicated open source tool that generates mobility traces readable in network simulators like NS2 [3] and Qualnet [4]. SUMO Player [5] and Trace Exporter [6] are also tools designed to convert netstate dump files (XML files containing all the simulation's details) generated from traffic microsimulators like SUMO (Simulation of Urban MObility), into mobility traces interpretable by network simulators like NS2 and VISSIM [7].

In the embedded approach, the traffic and the network simulators are natively coupled as they are incorporated to form a unique entity. MoVES [8], NCTUns [9] and VCOM [10] the embedded module in VISSIM fall into this category. However, the major limitation of these platforms is the simplified network or traffic capabilities. In fact, most of these platforms were initially meant either for traffic or network simulations but not both; and were provided by additional modules to support VANET simulations.

To overcome this limitation, the third category was introduced. In this case, a bidirectional communication is established between the simulators. The network simulator takes control of the traffic simulator by sending commands to change the nodes behavior. The latter responds to the former by reporting the positions of the loaded nodes. In the following,

this two-ways interaction will be referred to as *feedback loop*. This approach is considered so far as the most suitable solution in the literature for vehicular communications simulations offering an acceptable level of realism. TraNS [11], an open-source simulation environment, was the first implementing a federated approach; as it links the traffic simulator SUMO [12] and the network simulator NS2 using a specific interface called TraCI [13], which we will evoke further in Section 3. iTERIS [14] project, also an online approach implementation, aims at providing a framework to combine traffic mobility and network simulations for large scale testing of traffic management solutions. This framework uses SUMO as a traffic simulator and NS-3 [15] as a network simulator. Although it was a promising project, iTERIS has not been updated in its official website since 2010.

Ovnis [16] and Veins [17] are the latest attempts to implement an online simulation platform as it couples SUMO with NS-3, and SUMO with OMNeT++ [18] respectively.

The reason behind the miscarriage of the latest category, even though it presents the most realistic results one can get from the literature tools, is the long simulation times and the high computational cost. In fact, this constraint narrows the scope of the correspondent platforms to small scenarios, unless a supercomputer is used to perform the simulation. This will continue to restrain the progress of the ITS field as long as VANET simulations are not trivialized enough.

Our approach, named HINTS, advances all state-of-the-art solutions in terms of performance as it uses resources more efficiently and considerably reduces the computational cost and the simulation time. The combination of the online and the offline approaches brought forth a new approach that uses one full mobility trace and a feedback loop, for an active exchange and a real time interaction between the network and the traffic simulators.

III. HINTS DESCRIPTION

HINTS stands for Hybrid Integration of Network and Traffic Simulators. Our approach combines all the benefits from state of the art solutions. We propose an approach that is as flexible as the online approach, but with less computational cost and shorter simulations times by taking advantage of the offline simulation approach benefits also. As a result, our hybrid approach will be suitable for large scenarios simulations and will adapt with the number of events to offer better simulation times and less computational cost.

This approach is inspired from the hierarchical approach for image and video compression, where a first image is stored entirely as a background and the next images are obtained from storing the parts of the image that differ from the original one. We noticed many similarities between our problem and the image/video compression problem as both aim for reducing the computational cost and the size of data transferred in each time step, and reducing the size of data stored respectively. We noticed also that if an event occurs, this will be likely to alter the state of only a few nodes in the network while the bigger percentage of nodes will not be affected by this event. This led us to come up with a novel

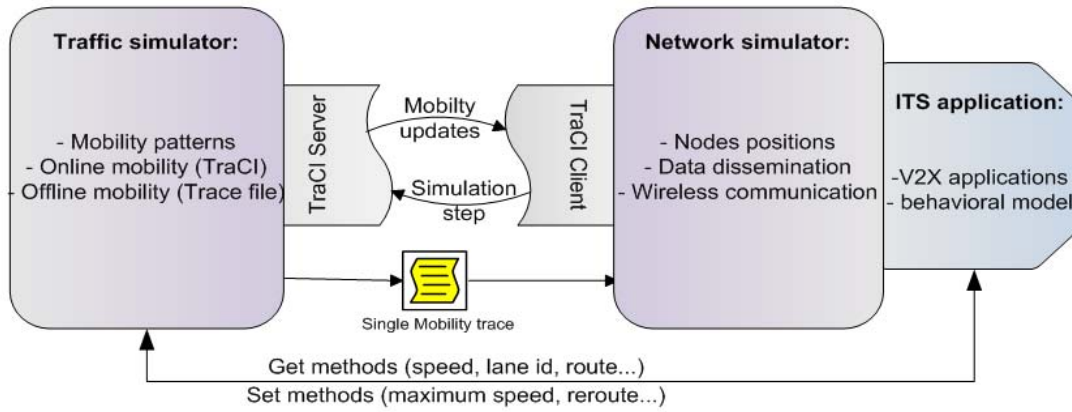


Figure 1. Platform general architecture

approach in this domain: the hybrid approach for realistic simulations of vehicular communications. In brief, we use a mobility trace as a background image of the network state, generated beforehand by the traffic simulator; and since the alteration in our case comes from the network simulator, we can determine exactly the nodes for which the behavior has changed. In this case, only the positions of the nodes concerned by an event are sent to the network simulator, and these new values will come to overload the values extracted from the original mobility trace.

IV. PLATFORM ARCHITECTURE

As pointed out in the first section, the analysis of protocols and applications in VANET require a real time bidirectional interaction between a network and a traffic simulator. HINTS consists of three essential blocks which are the traffic simulator, the network simulator and the ITS application. These three blocks interact with each other in real time via some dedicated communication interfaces. To benefit from the best tools in the literature, we chose the widely spread traffic simulator SUMO that has proved its reliability and performance within the Intelligent Transportation Systems community; and the successor of the most used network simulator: NS-3 also validated in the academia. The TraCI API accessible in SUMO has confirmed our selection for the traffic simulator as it allows running SUMO as a server and controlling it by any third-party application using a TCP socket connection. NS-3 was chosen due to its accurate implementation of physical layers and communications standards, its improved scalability (including parallelization capabilities) and its active development community. The accessibility of both simulators' source code in C++ is a further reason that reinforced our choice for these two simulators.

A. The general hybrid platform architecture

Our coupled platform uses SUMO in one hand to provide the required information about the state of the network (position, speed, lanes...) and supply the network simulator with the necessary feedback. In the other hand, NS-3 is used to

perform data dissemination in order to inform the ITS applications embedded on each node about the surrounding environment and the state of the network. It is now up to the ITS applications that takes the role of the behavioral model to decide whether a node should change its behavior or not. This efficient feedback loop is extended with the insertion of a mobility trace that includes the movements of all nodes from the entire simulation duration, and which represents a background image of the network; and thereby, considerably reduces the amount of data transferred between the traffic simulator and the network simulator in each time step since only the positions of nodes concerned by a certain event are sent.

Fig. 1 shows the general overview of the architecture. In this figure, the traffic simulator in the leftmost box manages the nodes' mobility and generates both static traces and dynamic micro-traces. The middle box represents the network simulator whose role is handling the data dissemination via wireless communications, in addition to translating received data about the nodes movements; and that, in order to implement its own realistic mobility model based on the decisions taken by the rightmost box, which represents the user level model. This latter block represents the communication protocol or user VANET application to be analyzed. The communication between these three blocks is provided by the TraCI client/server interface, which allows real time, bidirectional communication between the network and the traffic simulators.

B. Simulation flow overview

As the simulation starts, SUMO generates one complete mobility trace to constitute the base structure of the mobility model. This mobility trace is injected into the network simulator and the two simulators are launched simultaneously with NS-3 as the client application and SUMO as the server application. NS-3 reads from the previously generated mobility trace and sends commands to SUMO via TraCI to execute simulation steps in order to stay synchronized in time. Information about the surrounding environment of a node (speed, route, destination...) is provided by SUMO while the cooperative awareness messages (CAM) [19] exchanged between these nodes (safety messages, traffic state messages...) are provided by NS-3. All these information are

operated in the ITS application in order to decide whether a node should change its behavior (speed, trajectory, route or destination). Once the decision is taken for a node to change its

node, are themselves marked. At the end of each time step, the positions of all marked nodes are reported to the network simulator.

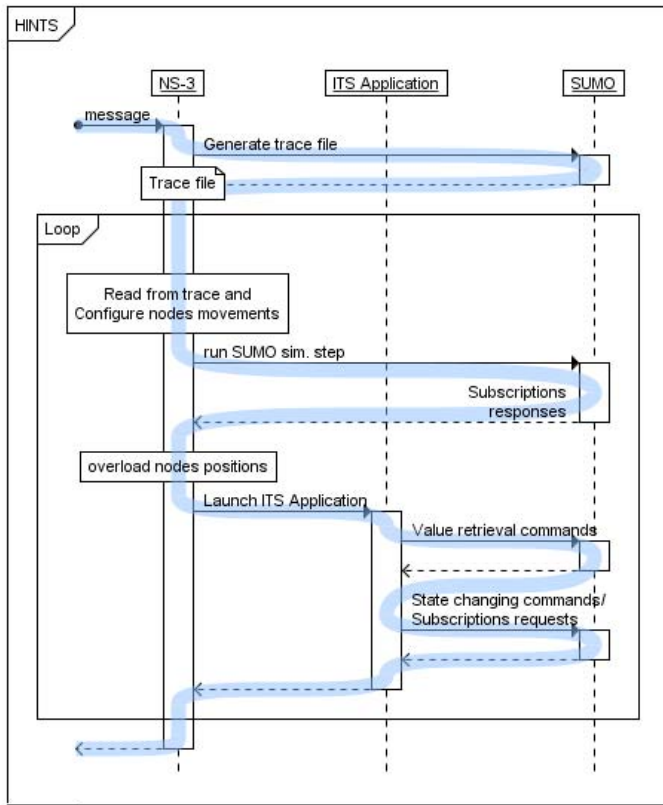


Figure 2. Sequence diagram in HINTS

trajectory, the corresponding information will be retrieved on each time step and until the node reaches its destination, and overload the values given by the static mobility trace. Fig. 2 shows the flow of events in HINTS.

One of the benefits of using this platform is that the mobility trace generated beforehand, can be reused multiple times with different scenarios of applications. In fact, analyzing an ITS application consists of defining a simulation scenario, testing it with different parameters to finally compare and evaluate its results. However, this will not be possible if we change the initial nodes mobility scenario each time we change a parameter. Another advantage of using this platform is the reduction of the amount of data sent and received between the two simulators. This will considerably shorten the simulation time and decrease the computational cost allowing simulating even larger scenarios.

C. SUMO and NS-3 implementation details

1) Modifications and additional modules

Note that the nodes in direct interaction with the nodes affected by a certain event are likely to change their trajectory too and the corresponding positions need, therefore, to be reported as well. To select these nodes, we used a simple technique that consists of marking each node interacting directly with an altered behavior node. In other words, all nodes in the same lane close enough from a marked node, and all nodes that want to change lane but are blocked by a marked

One inconvenient of the offline approach is the time taken to parse the generated mobility trace and convert it to a format understandable by the network simulator. For example, in the case of one simple intersection simulation, the generation of the raw output produced by SUMO for 1h30 scenario with 3400 nodes takes 10 seconds; while the conversion of this raw trace to a tcl format with two of the most efficient tools for this purpose, SUMO Player and Trace Exporter, takes 15 and 18 minutes respectively.

To tackle this shortcoming, we modified SUMO so it produces an optimized generic format of mobility traces (tcl format) directly usable in NS-3. This way, the mobility trace generation is shortened to the traffic microsimulation time, which is relatively short. In addition to that, we have optimized NS-3 so it reads through the produced mobility trace more efficiently by preventing it from parsing the file multiple times, and reading only the desired portions of the trace instead, at each time step, allowing gaining considerable time.

To furnish our platform with an efficient feedback loop (Fig. 1), we have provided NS-3 with a TraCI client that takes control of the SUMO simulation via the TraCI server, and therefore, supplies NS-3 and the ITS application with the needed data about the network and the nodes mobility; with the help of value retrieval methods, state changing methods and control-related methods. The latter category performs SUMO simulation steps, and subscriptions of the concerned nodes (speed and value retrieval on each time step).

2) Specifications about HINTS

We have classified the events that are susceptible to alter the behavior of a node into two essential classes: primary events and secondary events. The first class touches the infrastructure and affects the mobility of nodes in a major way, while the second class only concerns a group of vehicles, and affects the traffic in a more microscopic manner. The reason behind this appellation is due to the nature of the events. In other words, the occurrence of a primary event will lead to the occurrence of multiple secondary events. As an example for a primary event, we can cite a blocked road. This event will most probably lead to traffic congestion and a number of vehicles changing their routes, which in this case, are considered as secondary events. Based on this classification and the VANET application taxonomy proposed by the C2C-CC [20], we decided to provide the TraCI client with five vehicle state changing mobility primitives: Stop, Change speed, Change lane, Change route and Change target. These five primitives are sufficient to handle mobility alteration in a microscopic level from any scenario, and thereby, increase the efficiency of our feedback loop.

HINTS was designed to reduce the simulation complexity procedure from a developer point of view. Apart from the XML files needed to generate the mobility trace in SUMO, and the ITS application written in C++ and executed in NS-3; no additional files will be required for the simulation execution.

Also, most of this simulation process will be automated. From NS-3 side, a simple call for the method “Install ()” which configures the positions of nodes, will trigger the synchronized execution of SUMO time steps. In addition to that, subscription messages are sent to SUMO automatically with each call for a state changing method of the corresponding node. The novelty for the developer is the use of the methods provided by our platform to supply the ITS application with the required data (get speed, get lane, get route...) and influence the mobility model (stop node, set speed, reroute...) as wished.

V. PERFORMANCE EVALUATION

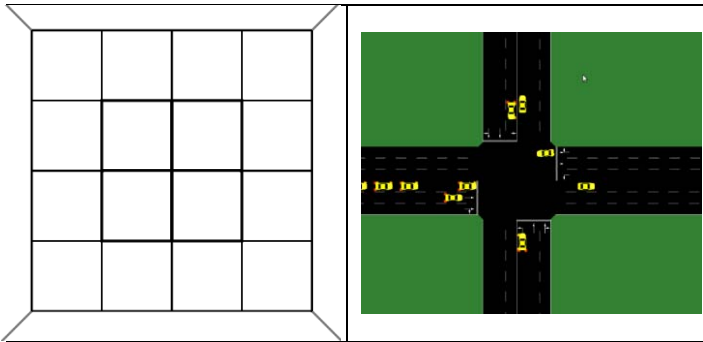


Figure 3. Topology

In order to prove the efficiency and the performance of the developed platform, we considered testing it on a modified Manhattan grid scenario depicted in Fig. 3 and representing a city of 20000 inhabitants. We used a 2 km² network with 3 types of roads (1 lane, 2 lanes and 3 lanes) to show the importance of each road. The represented city has 4 main entrances. In the following, we will briefly introduce the operating mode of the tested ITS application. The performance of HINTS is justified using the percentage of subscriptions metric while the efficiency is substantiated using a metric called Mean Sojourn Time or the Mean Travel Time metric in our case: the average time vehicles spend on the road before reaching their destinations, calculated for the arrived vehicles at the end of each simulation step.

A. Application’s operating mode

In the following, we assume that vehicles are equipped with interfaces for wireless communications, and also with an onboard GPS that keeps track of the current road id. Each vehicle follows its planned route; if any unexpected event occurs (accident, blocked road), the vehicle in front will emit warning messages within a range of 250 meters. This warning message is composed of the current road id and time. Vehicles that receive the warning message can then check if this road is part of the initially planned route with the help of the onboard GPS, and can then change their route if possible. The following algorithm shows the iterative behavior of the application embedded on vehicles:

```

1: if (stop_condition) then
2:   StopApplication();
3: end if
4:   current_road ← GetRoadId();
5: if (road_is_blocked) then
6:   BroadcastPacket(time , current_road);
7: end if
8: if (ReceivePacket(time, roadId)) then
9:   if (current_road = roadId) then
10:    ChangeRoute();
11:   end if
12: end if

```

B. Results

The simulated scenario counts an overall of 7730 vehicles running for 5400 seconds. The simulation starts at time 25000, which corresponds to the morning rush hour (around 7:00 AM). At time 26000 seconds, we block one of the main roads of the city. The road remains blocked for 2000 seconds then it is opened again. In Fig. 4, we show the performance of our platform by illustrating the decreased number of subscribed vehicles (vehicles for whom the speed and positions are retrieved from SUMO on each time step).

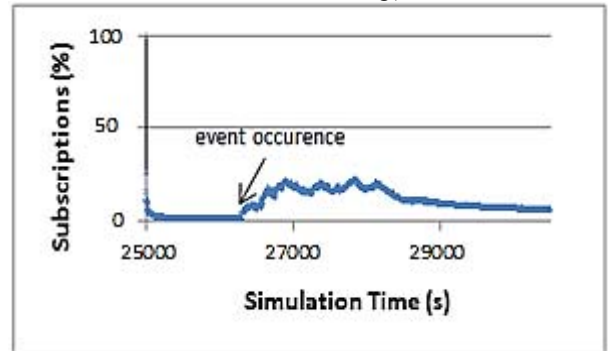
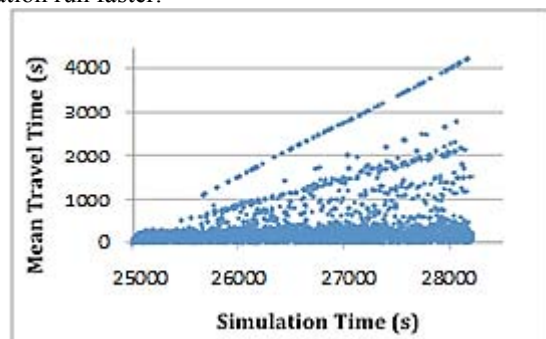
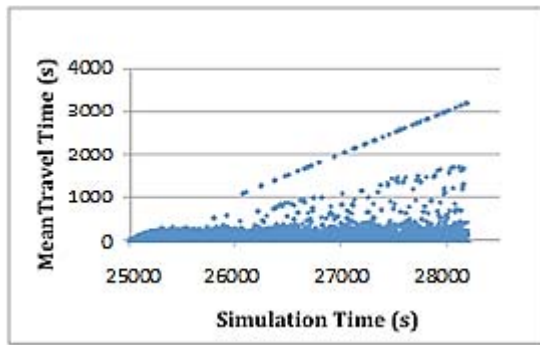


Figure 4. Percentage of the subscribed vehicles

We can notice that the number of subscriptions doesn’t exceed 25% of the loaded vehicles over the event occurrence period; while an online implementation requires the subscriptions of 100% of the loaded vehicles. This allows decreasing the simulations’ computational cost letting the simulation run faster.



(a)



(b)

Figure 5. Mean travel time (a) without the application, and (b) with the application

The efficiency of our platform is shown using the Mean Travel Time metric. The results reflecting the fluidity of road traffic obtained after launching the tested application are illustrated in Fig. 4. The minimum interval between two packets sent by a vehicle was set to 5 seconds and the time to live for a packet is 1 second. In the observed period, the number of vehicles and the mean travel times increases starting from around 25600 seconds. Fig. 5 (a) shows the mean travel time of vehicles without the application reaching higher than 4000s in some cases. The enhancement can be observed in Fig. 5 (b), showing the mean travel times with the application launched, as the maximum travel time is slightly higher than 3000s. In other terms, vehicles spend less time on the road and reach their destinations faster compared with the scenario wanting the application.

VI. CONCLUSION

In this paper, we introduced a novel approach for realistic simulations of vehicular communications, and the design of the platform HINTS, the corresponding implementation. It reposes on SUMO for the realistic mobility of nodes, and on NS-3 for the wireless communications. The goal of this new approach is to decrease the simulations' times and their computational costs in order to trivialize VANET application simulations. Future works will focus on improving the open source platform HINTS, and using it for its main purpose, that is, to evaluate VANETs protocols and applications.

REFERENCES

[1] H. Hartenstein, K.P. Laberteaux "VANET : Vehicular Applications and Inter-Networking Technologies".
 [2] "MOVE [online], omla_version/ http://lens.csie.ncku.edu.tw/Jo[online], index.php/research- projects/past/18-rapid-vanet

[3] "The network simulator ns-2 [online]," http://www.isi.edu/nsnam/ns/
 [4] "Qualnet [online]," http://www.scalable-networks.com
 [5] "SUMOPlayer [online]" http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Contributed/SUMOPlayer
 [6] "TraceExporter [online]" http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Tools/TraceExporter
 [7] C. Schroth, F. Dötzer, T. Kosch, B. Ostermaier and M. Strassberger, "Simulating the traffic effects of vehicle-to-vehicle messaging systems" 5th International Conference on ITS Telecommunications (ITS-T) 2005.
 [8] L. Bononi, M. Di Felice, G. D'Angelo, M. Bracuto, L. Donatiello "MoVES: A framework for parallel and distributed simulation of wireless vehicular ad hoc networks".
 [9] S.Y. Wang, C.C Lin "NCTUns 5.0: A Network Simulator for IEEE 802.11(p) and 1609 Wireless Vehicular Network Researches" VTC Fall 2008.
 [10] M. Killat, F. Schmidt-Eisenlohr, H. Hartenstein, C. Rössel, P. Vortisch, S. Assenmacher, F. Busch, "Enabling efficient and accurate large-scale simulations of VANETs for vehicular traffic management" in VANET '07 Proceedings of the fourth ACM international workshop on Vehicular ad hoc networksPages 29-38.
 [11] M. Pi 'orkowski, M. Raya, A.L. Lugo et all "TraNS: Realistic Joint Traffic and Network Simulator for VANETs". ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 12, Nr. 1, pp. 31--33, 2008.
 [12] "Simulation of urban mobility [online]," http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Main Page
 [13] A.Wegener, M.P.'orkowski, M Raya, H. Hellbrück, S. Fischer and J-P . Hubaux "TraCI: An Interface for Coupling Road Traffic and Network Simulators" in CNS '08 Proceedings of the 11th communications and networking simulation symposium Pages 155-163
 [14] J. Gozalvez, S. Turksma, L Lan, O. Lazaro et all "iTETRIS: the Framework for Large-Scale Research on the Impact of Cooperative Wireless Vehicular Communications Systems in Traffic Efficiency". in ICT-MobileSummit , 10-12 June 2009 .
 [15] "The ns-3 network simulator [online]," http://www.nsnam.org/
 [16] Y. Pigné "A Platform for Realistic Online Vehicular Network Management". in IEEE International Workshop on Management of Emerging Networks and Services 2010.
 [17] C. Sommer, F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis". in IEEE Transactions on Mobile Computing 10 (2011), Nr. 1, S. 3-15.
 [18] "Omnet++ [online]," http://www.omnetpp.org/.
 [19] ETSI TS 102637-2 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service
 [20] "CAR 2 CAR Communication Consortium [online]" http://www.car-to-car.org/
 [21] "Ovnis [online]," http://ovnis.gforge.uni.lu
 [22] "TraNS [online]," http://trans.epfl.ch/