2015 29th International Conference on Advanced Information Networking and Applications Workshops

Performance Evaluation of a VANET Simulation System Using NS-3 and SUMO

Vladi Kolici*, Tetsuya Oda†, Evjola Spaho*, Leonard Barolli‡, Makoto Ikeda‡ and Kazunori Uchida‡

**Faculty of Information Technologies*
Polytechnic University of Tirana
Bul. "Dëshmorët e Kombit", "Mother Theresa" Square, Nr. 4, Tirana, Albania
E-mail: vkolici@fti.edu.al, evjolaspaho@hotmail.com

†*Graduate School of Engineering,*
Fukuoka Institute of Technology (FIT),
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811–0295, Japan
Email: oda.tetsuya.fit@gmail.com

‡*Department of Information and Communication Engineering,*
Fukuoka Institute of Technology (FIT),
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811–0295, Japan
Email: barolli@fit.ac.jp, makoto.ikd@acm.org, k-uchida@fit.ac.jp

***Abstract*—In this paper, we investigate the performance of HWMP, OLSR and DD protocols in a VANET crossroad scenario. The mobility patterns of vehicles are generated by means of SUMO (Simulation of Urban MObility) and as communication protocol simulator is used NS3 (Network Simulator 3). For the simulations, we used IEEE 802.11p standard, TwoRayGround Propagation Loss Model and sent multiple CBR flows over UDP between 20 source-destination pairs. We use Packet Delivery Ratio (PDR), throughput and delay as evaluation metrics. We compared the performance of three protocols and the simulation results shows that the DD protocol has better PDR and throughput compared with HWMP and OLSR protocol. However, the DD protocol has a long delay because is a delay tolerant protocol.**

***Keywords*-Vehicular Ad-Hoc Networks, VANET, NS3, SUMO.**

I. INTRODUCTION

Vehicular Ad-hoc Networks (VANETs) are a special type of Ad-hoc networks. They can been utilized to guarantee road safety, to avoid potential accidents and make new forms of inter-vehicle communications so they will be an important part of the future Intelligent Transport Systems (ITS).

Due to the high cost of deploying and implementing VANET systems in a real environment, most of research is concentrated on simulations.

In the recent years, a lot of simulators for VANETs have been emerging [1]. For example, the IMPORTANT framework has been one of the first attempt to understand the dependence between vehicular traffic and communication performance [2], [3]. The authors analyzed the impact of the node mobility on the duration of communication paths.

In [4], the authors present a simulator written in Java, which can generate mobility traces in several formats. There are also other traffic simulators, like TranSim [5], which makes use of a cellular automaton for simulating the interaction of vehicles. Cellular Automaton Based Vehicular Network (CAVENET) [6], [7] is a lightweight simulator written in MATLAB and ns-2/ns-3 which can be used to understand the properties of the mobility models of vehicular traffic and their impact on the performance of VANETs.

The Opportunistic Network Environment (ONE) [8] is a simulation environment that is capable of generating node movement using different movement models, routing messages between nodes with various DTN routing algorithms and sender and receiver types, visualizing both mobility and message passing in real time in its graphical user interface. ONE can import mobility data from real-world traces or other mobility generators. It can also produce a variety of reports from node movement to message passing and general statistics.

Also, SUMO is another traffic simulator, intended for traffic planning and road design optimization. There is an attempt to interface SUMO with NS2 [9]. In this regard, we used SUMO simulator which can be used to understand the properties of the mobility models of vehicular traffic and their impact on the performance of VANETs and interface it with NS3 to evaluate the performance of routing protocols.

In our previous work [10], we evaluated the performance of three routing protocols: Ad-hoc On-demand Distance Vector (AODV) [11], Optimized Link State Routing (OLSR) [12] and Dynamic MANET On Demand (DYMO) [13], [14], using NS2 network simulator [15].

In this work, we use NS3 to evaluate the performance of three routing protocols (HWMP, OLSR and DD) in a VANET crossroad scenario.

The rest of the paper is structured as follows. In Section II, we describe different mobility models in VANETs. In

978-1-4799-1775-4/15 $31.00 © 2015 IEEE
DOI 10.1109/WAINA.2015.121

Section III, we briefly describe HWMP, OLSR and DD protocols. The simulation system design and description is presented in Section IV. In Section V, we show the simulation results. Finally, the conclusions and future work are presented in Section VI.

II. Mobility Models in VANETs

A Random Waypoint (RW) model has been the earliest mobility model for ad-hoc networks. Basically, in RW every node picks up a random destination and a random velocity at certain points called waypoints. This model has been extended in a number of ways in order to take into account more realistic movements.

The RW model, the Random Walk model, the Random Direction model, the Reference Point Group (or Platoon) model, the Node Following mode, the Gauss-Markov model, all involved generation of random linear speed-constant movements within the topology boundaries.

The Random Waypoint City Model [16] approach combines aspects of the RW Mobility Model with the vector street maps. It has a high granularity as exact user locations are available. User movement is independent of other users and past trips, so that individual homes and workplaces of users are not modelled.

STRAW (STreet RAndom Waypoint) [17] is another mobility model for VANETs that 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. This mobility model provides reasonable runtimes and memory consumption that scales fairly well with the size of the simulation.

Manhattan model is a generated-map-based model introduced in [2] to simulate a urban environment. The simulation area is represented by a map (generated before the simulation start) containing vertical and horizontal roads made up of two lanes, allowing the motion in the two directions (north-south for the vertical roads and east-west for the horizontal ones). Contrary to the freeway model, a vehicle can change a lane when it passes a crossroads with absolutely no control mechanism, thus continuing their movements without stopping. This makes this model unrealistic.

In City Section Mobility Model, the simulation area is a street network that represents a section of a city where the ad hoc network exists. The streets and speed limits are based on the type of city being simulated. This model can be seen as a hybrid model between RWP and Manhattan, as it introduces the principle of RWP like the pause-time and random selection destination, within a generated-map-based area. This model is only for small simulation area.

Stop Sign Model [18] is the first model that integrates a traffic control mechanism. This model is based on real maps of the TIGER/Lines database, but all roads are assigned a single lane in each direction and a vehicle should never overtake its successor. In the Traffic Sign Model model, stop signals are replaced by traffic lights. A vehicle stops at a crossroads when it encounters a red stoplight, otherwise it continues its movement. When the first vehicle reaches the intersection, the light is randomly turned red with probability p (thus turned green with probability $1 - p$). If it turns red it remains so for a random delay (pause-time), forcing the vehicle to stop as well as the ones behind it.

III. Routing Protocols

For our performance comparison study, we choose different routing protocols HWMP, OLSR and DD. We will shortly describe these protocols in following.

A. HWMP Protocol

Hybrid Wireless Mesh Protocol (HWMP) defined in IEEE 802.11s, is a basic routing protocol for a wireless mesh network. It is based on AODV [19] and tree-based routing. It relies on peer link management protocol by which each mesh point discovers and tracks neighboring nodes. If any of these are connected to a wired backhaul, there is no need for HWMP, which selects paths from those assembled by compiling all mesh point peers into one composite map.

HWMP protocol is hybrid, because it supports two kinds of path selection protocols. Although these protocols are very similar to routing protocols in case of IEEE 802.11s they use MAC addresses for "routing", instead of IP addresses. Therefore, we use the term "path" instead of "route" and thus "path selection" instead of "routing".

HWMP is intended to displace proprietary protocols used by vendors like Meraki for the same purpose, permitting peer participation by open source router firmware.

B. OLSR Protocol

The OLSR protocol [12] is a pro-active routing protocol, which builds up a route for data transmission by maintaining a routing table inside every node of the network. The routing table is computed upon the knowledge of topology information, which is exchanged by means of Topology Control (TC) packets.

OLSR makes use of `HELLO` messages to find its one hop neighbours and its two hop neighbours through their responses. The sender can then select its Multi Point Relays (MPR) based on the one hop node which offer the best routes to the two hop nodes. By this way, the amount of control traffic can be reduced. Each node has also an MPR selector set which enumerates nodes that have selected it as an MPR node. OLSR uses TC messages along with MPR forwarding to disseminate neighbour information throughout the network. Host Network Address (HNA) messages are used by OLSR to disseminate network route advertisements in the same way TC messages advertise host routes.

C. DD protocol

The core of the DD protocol is based on the concept of a utility function. A utility function assigns a utility value, U_i, to every packet i, which is based on the metric being optimized. The U_i is defined as the expected contribution of packet i to this metric. The DD replicates packets first that locally result in the highest increase in utility. For example, assume the metric to optimize is average delay. The utility function defined for average delay is $U_i = -D(i)$, basically the negative of the average delay. Hence, the protocol replicates the packet that results in the greatest decrease in delay. The DD, like MaxProp, is flooding-based, and will therefore attempt to replicate all packets if network resources allow.

The overall protocol is composed of four steps:

1) Initialization: Metadata is exchanged to help estimate packet utilities.
2) Direct Delivery: Packets destined for immediate neighbors are transmitted.
3) Replication: Packets are replicated based on marginal utility (the change is utility over the size of the packet).
4) Termination: The protocol ends when contacts break or all packets have been replicated.

IV. Simulation System Design and Description

A. Simulation System Structure

The simulation system structure is shown in Fig. 1. The behavioural analyzer block (SUMO), generates the movement pattern of the vehicles that is used by the communication protocol analyzer (NS3 simulator).

The detailed simulation model is based on NS-3 (version 3.14.1) [20]. The NS3 simulator is developed and distributed completely in the C++ programming language, because it better facilitated the inclusion of C-based implementation code. The NS3 architecture is similar to Linux computers, with internal interface and application interfaces such as network interfaces, device drivers and sockets. The goals of NS3 are set very high: to create a new network simulator aligned with modern research needs and develop it in an open source community. Users of NS3 are free to write their simulation scripts as either C++ main() programs or Python programs. The NS3 low-level API is oriented towards the power-user but more accessible "helper" APIs are overlaid on top of the low-level API.

In order to achieve scalability of a very large number of simulated network elements, the NS3 simulation tools also support distributed simulation. The NS3 support standardized output formats for trace data, such as the pcap format used by network packet analyzing. tools such as tcpdump, and a standardized input format such as importing mobility trace files from NS2 [15].

The NS3 simulator has models for all network elements that comprise a computer network. For example, network devices represent the physical device that connects a node to the communication channel. This might be a simple Ethernet network interface card, or a more complex wireless IEEE 802.11 device.

Table I
SIMULATION PARAMETERS.

Parameters	Values
Network Simulator	NS3
Traffic Simulator	SUMO
Number of vehcles	20, 110, 220
Mapmodel	Watanabedori-1chome
PHY	IEEE 802.11p
Propagation loss model	Log-distance Propagation Loss Model
Propagation delay model	Constant Speed Model
Routing protocol	HWMP, OLSR, DD
Transport protocol	UDP
Application type	CBR
Packet size	512 [Bytes]
Transmission rate	512 [kbps]
Number of flows	20
Simulation time	660 [sec]

In our simulations we used IEEE 802.11p standard and TwoRayGround Propagation Loss Model.

IEEE 802.11p: It is an approved amendment to the IEEE 802.11 standard to add wireless access in vehicular environments (WAVE). It defines enhancements to 802.11 required to support ITS applications. The 802.11p standard is based on the 802.11 architecture, but version "p" is aimed at communications between vehicles and between them and fixed infrastructure. This new technology uses the 5.9 GHz band in various propagation environments: vehicle, open, urban, and so on. This standard defines the WAVE as the signalling technique and interface functions that are controlled by the physical layer (MAC) devices where the physical layer properties change rapidly and where the exchanges of information have a short duration. The purpose of this standard is to provide a set of specifications to ensure interoperability between wireless devices trying to communicate in rapidly changing environments and in particular time periods.

TwoRayGroundPropagationLossModel: It considers the direct path and a ground reflection path. The received power at distance t is calculated with the following equation:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L}.$$

where h_t and h_r are the heights of the transmit and receive antennas.

Simulation Environment: The simulations are carried out in Ubuntu 11.10.

Traffic Model: Constant Bit Rate (CBR) traffic sources and 512 bytes data packets are used. Other simulation parameters are shown in Table 1.

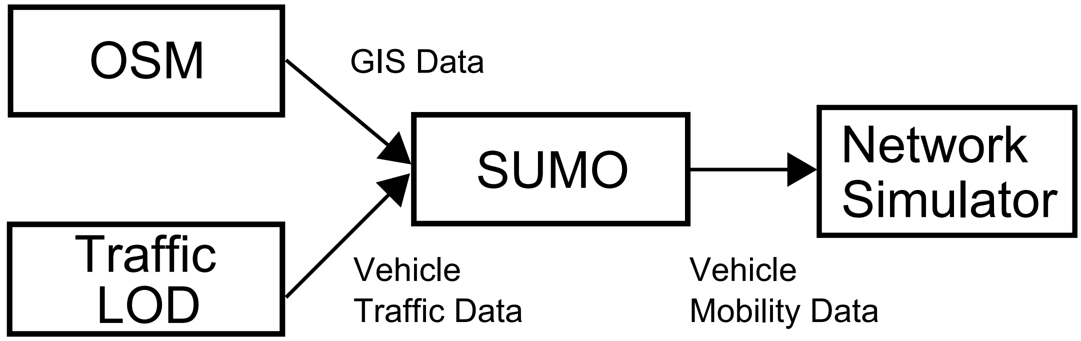


Figure 1. Simulation system structure.

B. Implementation of a Crossroad Scenario with SUMO and Simulation Description

In this work, we consider two lanes crossroad scenario (see Fig. 2). During the simulations, 20, 110, 220 vehicles move in roads according to RW model by respecting the traffic light. Vehicles are randomly positioned at the begin of the intersection. The maximum speed of the vehicles is 135 km/h. The vehicles accelerate and decelerate their speed. The vehicles follow each other and when a vehicle reach at the end of the road it is shifted at the begin of the same road. When a vehicle goes near the intersection, it checks for the state of the traffic light. The traffic lights change the state in a constant time period. If the traffic light is red, it decelerates and stop. If the traffic light is green it reduces the speed and enter to the intersection. The cars in front of the crossroad and in the junction will have a speed of 27 km/h (minimal).

For simulations, we used HWMP, OLSR and DD protocols and sent multiple CBR flows over UDP. Ten connections are created and the same source-destination pairs are used for both protocols. The simulation time is 660 sec.

V. Simulation Results

We present here some simulation results for HWMP, OLSR and DD protocols done by means of SUMO and NS3. For performance evaluation, we use three metrics: the average PDR, throughput and delay.

In Fig. 3 are shown the simulation results of PDR for three protocols. We can see that the performance of DD protocol is better than HWMP and OLSR protocols.

In Fig. 4 are shown the simulation results of throughput. For small number of vehicles, the HWMP protocol performs better than other protocols. However, when the number of vehicles is 220, the DD protocol has a better performance than HWMP and OLSR protocols.

The simulation results for delay are shown in Fig. 5. The delay of DD protocol is higher than other protocols, because DD is a delay tolerant protocol. For other protocols, with the increase of the number of vehicles, the delay is decreased. This is because the density of the nodes is increased and the communication between nodes becomes better.

VI. Conclusions

In this work, we evaluated the performance of HWMP, OLSR and DD routing protocols in a VANET crossroad scenario considering PDR, throughput and delay as evaluation metrics. We used SUMO to generate the movement of the vehicles and NS3 to test the performance of routing protocols. For simulations, we considered IEEE 802.11p standard and TwoRayGround Propagation Loss Model and sent multiple CBR flows over UDP. We also considered different number of vehicles (20, 110 and 220 vehicles). From simulations, we found the following results.

- Considering PDR, the performance of DD protocol is better than HWMP and OLSR protocols.
- For small number of vehicles, the throughput of HWMP protocol is better than other protocols. However, when the number of vehicles is 220, the DD protocol has a better performance than HWMP and OLSR protocols.
- The delay of DD protocol is higher than other protocols, because DD is a delay tolerant protocol. For other protocols, with the increase of the number of vehicles, the delay is decreased. This is because the density of the nodes is increased and the communication between nodes becomes better.

In the future, we plan to extend our work for different radio propagation models and environments. We also would like to consider other parameters for evaluation such as routing overhead, traffic quantity and topology change.

References

[1] J. Harri, F. Filali and C. Bonnet, *"Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy"*, IEEE Communications Surveys & Tutorials, Vol. 11, No. 4, pp. 19-41, 2009.

[2] F. Bai, N. Sadagopan, and A. Helmy, *"IMPORTANT: A Framework to Systematically Analyze the Impact of Mobility on Performance of Routing Protocols for Ad-hoc Networks"*, In Proc. of IEEE INFOCOM-2003, pp. 825-835, March-April 2003.

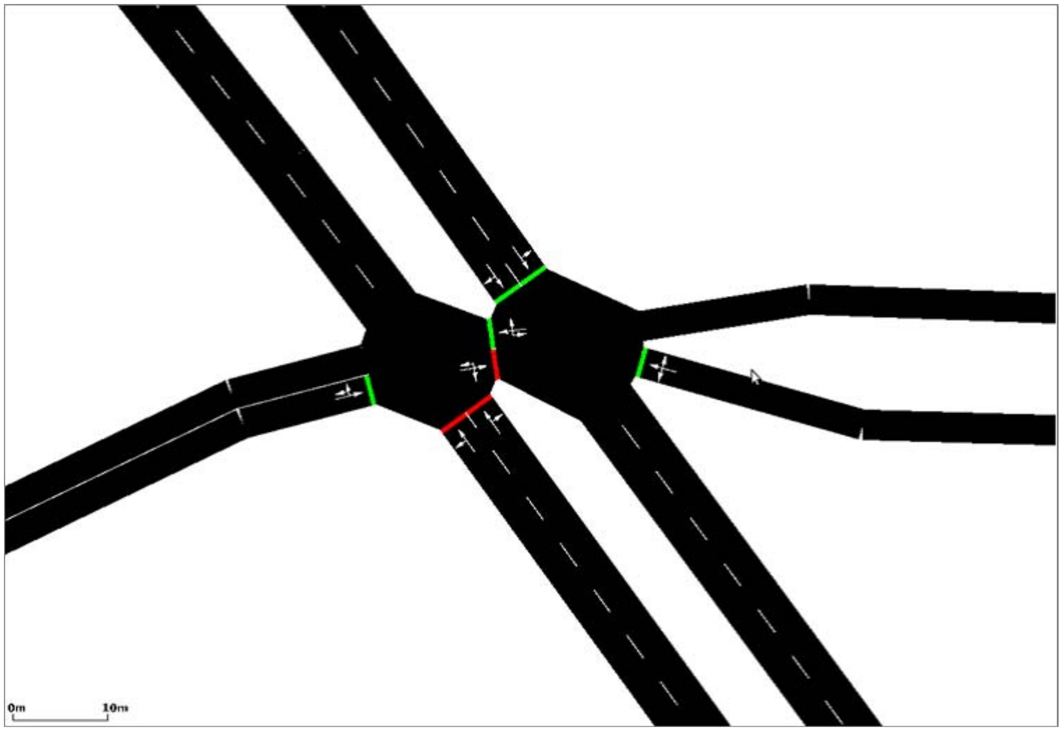
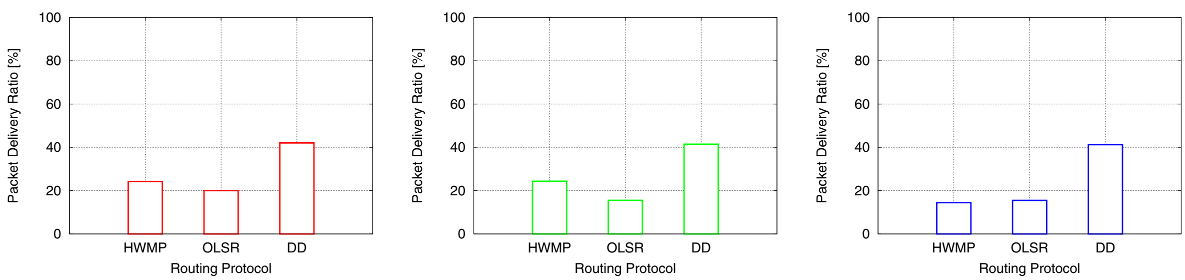


Figure 2. Simulation scenario.

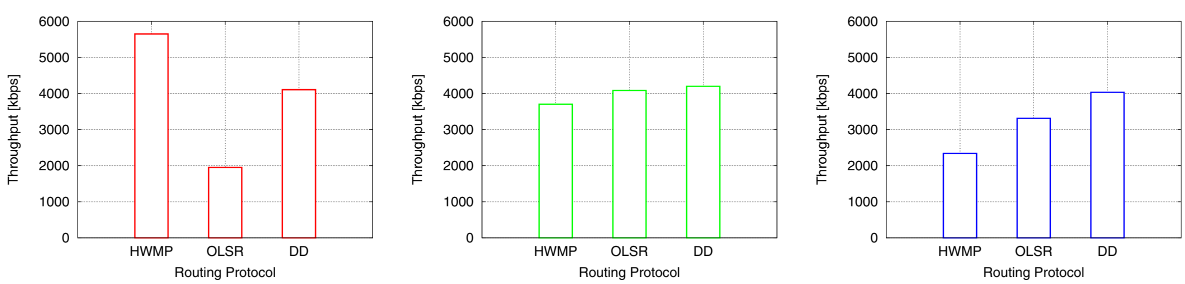


(a) Number of Vehicles: 20

(b) Number of Vehicles: 110

(c) Number of Vehicles: 220

Figure 3. Simulation Results: PDR

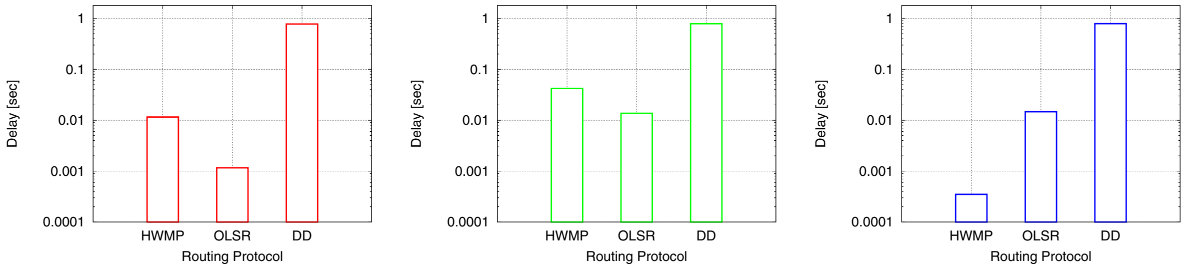


(a) Number of Vehicles: 20

(b) Number of Vehicles: 110

(c) Number of Vehicles: 220

Figure 4. Simulation Results: Throughput



(a) Number of Vehicles: 20

(b) Number of Vehicles: 110

(c) Number of Vehicles: 220

Figure 5. Simulation Results: Delay

[3] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy, *"PATHS: Analysis of Path Duration Statistics and Their Impact on Reactive MANET Routing Protocols"*, In Proc. of the 4-th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc-2003), pp. 245-256, 2003.

[4] M. Fiore, J. Harri, F. Filali, and C. Bonnet, *"Vehicular Mobility Simulation for VANETs"*, Proc. of the 40-th Annual Simulation Symposium (ANSS-2007), pp. 301-309, 2007.

[5] L. Smith, R. Beckan, R. Anson, K. Nagel, and M. Williams, *"TRANSIMS: Transportation Analysis and Simulation System"*, In Proc. of the 5-th National Transportation Planning Methods Applications Conference, LA-UR 95-1664, April 1995.

[6] G. De Marco, M. Tadauchi and L. Barolli, *"Description and Analysis of a Toolbox for Vehicular Networks Simulation"*, In Proc. of IEEE ICPADS/PMAC-2WN-2007, Vol. 2, pp. 1-6, 2007.

[7] K. Nagel and M. Schreckenberg, *"A Cellular Automaton Model for Freeway Traffic"*, Journal of Physics I France, Vol. 2, pp. 2221-2229, 1992.

[8] A. Keranen, J. Ott, and T. Karkkainen, "The ONE Simulator for DTN Protocol Evaluation", In Proc. of the 2-nd International Conference on Simulation Tools and Techniques (SIMUTools-2009), 2009, http://www.netlab.tkk.fi/tutkimus/dtn/theone/pub/the_one_simutools.pdf.

[9] M. Piorkowski, M. Raya, A. L. Lugo, M. Grossglauser, and J. P. Hubaux, *"Joint Traffic and Network Simulator for VANETs"*, In Proc. of Mobile and Information Communication Systems Conference (MICS-2006), October 2006, Available on line at: http://www.mics.ch/.

[10] E. Spaho, Gj. Mino, L. Barolli and F. Xhafa, *"Goodput and PDR Analysis of AODV, OLSR and DYMO Protocols for Vehicular Networks using CAVENET"*, International Journal of Grid and Utility Computing (IJGUC), Vol. 2, No. 2, pp. 130-138, 2011.

[11] C. Perkins, E. Belding-Royer and S. Das, *"Ad hoc On-Demand Distance Vector (AODV) Routing"*, IETF RFC 3561 (Experimental), 2003.

[12] T. Clausen and P. Jacquet, *"Optimized Link State Routing Protocol (OLSR)"*, IETF RFC 3626, October 2003.

[13] I. Chakeres and C. Perkins, *"Dynamic MANET On-demand (DYMO) Routing"*, Internet Draft (draft-ietf-manet-dymo-14), 2008, Work in progress.

[14] C. Sommer and F. Dressler, *"The DYMO Routing Protocol in VANET Scenarios"*, In Proc. of 66-th IEEE Vehicular Technology Conference (VTC2007-Fall), pp. 16-20, 2007.

[15] Information Science Institute (ISI), *"Network Simulator Version 2 (NS-2)"*, http://www.isi.edu/nsnam.

[16] J. Kraaier, U. Killat, *"The Random Waypoint City Model, User Distribution in a Street-Based Mobility Model for Wireless Network Simulations"*, In Proc. of WMASH-2005, Cologne, Germany, pp. 100-103, 2005.

[17] D. Choffnes, F. Bustamante, *"STRAW: An Integrated Mobility and Traffic Model for VANETs"*, In Proc. of the 10th International Command and Control Research and Technology Symposium (CCRTS-2005), pp. 69-78, 2005.

[18] N. Potnis and A. Mahajan,*"Mobility Models for Vehicular Ad hoc Network Simulations"*, In Proc. of the 44th annual Southeast Regional Conference, New York, pp. 746-747, 2006.

[19] C. Perkins, E. Belding-Royer, and S. Das, gAd hoc On-Demand Distance Vector (AODV) Routing,h RFC 3561 (Experimental), July 2003.

[20] *"The ns-3 Network Simulator"*, http://www.nsnam.org.