Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Simulation Tools in Vehicular Adhoc Network: A Challenge

[1]Mallika Gandhi, [2]Dr. Ashish Gupta
[1]Phd (CSE), Sharda University and Assistant Professor, ITS Engineering College, Uttar Pradesh, India
[2]Professor, Department of Computer Science & Engineering, ITS Engineering College, Uttar Pradesh, India

Abstract - Wireless communication technologies have impacted our daily lives. Vehicular Ad Hoc Networks (VANETS) is a type of wireless network or we can say special type of MANET (mobile ad hoc networks) in which the nodes can be vehicles available on the road which can move in any direction with varying speed. This irregular motion of vehicles develops new challenges for researchers and developers in terms of designing robust protocols for VANETS. Researchers perform various tests through simulation using various metrics to check how VANETS perform, before they are used in commercial application in the real world. VANET simulation is basically different from other type of wireless network MANETs simulation because in VANETs, vehicular environment poses new concerns and requirements, such as multi-path fading, constrained road topology and roadside obstacles, trip models, varying vehicular speed and mobility, traffic lights, traffic congestion, drivers' behavior, , traffic flow models etc. Currently, VANET simulation can be classified as VANET network simulators, mobility generators, and VANET simulators. This paper covers a review of different VANET simulation software and their components publicly available and the challenges in implementing those simulations.

Keyword - VANET, mobility generators,network simulators, VANET simulators.

I. INTRODUCTION

VANET has become major potential technology in the last few years to enhance safety as well as travel comfort. Many fortuitous situations are faced on the roads daily, which may lead to congestion and chaos. If vehicles or nodes in the network could be provided with some useful information before hand about such incidents or traffic conditions, the quality of driving can be improved significantly in terms of safety, time and distance. One of the major issues in VANET is of finding and maintaining an effective route for transporting data information. Security and privacy play an essential role in vehicular communications for successful acceptance and deployment of such a technology. The vehicular safety application should be completely tested before it is deployed in a real world to use. Simulator tool has been preferred over real world experiment because it less complex and economical. VANET requires that a traffic and network simulator should be used together to perform this test [1].

Simulation in VANET is a challenge because analysis of VANET applications requires that both a vehicle motion and a data network simulator can be used at the same time i.e. feeding simulation data to each other and deploying and testing of VANETs in real world is expensive and requires intensive labor. Therefore simulation tools are a useful alternative prior to actual implementation. This paper discusses major issues and challenges in deploying protocols on different simulators [2, 3].

II. SIMULATION TOOLS CLASSIFICATION

VANET simulation software can be classified as:

Vehicle Mobility Generators: are used to increase the level of realism in VANET simulations as they generate realistic vehicular mobility traces to be used as an input for a network simulator. The inputs to the mobility generator are the road model, scenario parameters (i.e., maximum vehicular speed, rates of vehicle arrivals and departures, etc). The output of mobility generators are the trace details i.e. the location of each vehicle at every time instant for the entire simulation time and their mobility profiles. Different examples are SUMO [11], MOVE [12], CityMob [13], STRAW[14], FreeSim [15], Netstream [16], and VanetMobiSim [17]. There are two types of Vehicle Mobility Generators:

Macro-Mobility: refers 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 to microscopic aspects which influence vehicular traffic, 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 is important for a reliable VANET simulation that both macro-mobility and micro-mobility descriptions are both considered when modeling vehicular movements. Models that have been widely used within the traffic science community include the *Cellular Automaton* (CA) model [20], the *Stefan Krauss* (SK) model [31], and the *Intelligent Driving Model* (IDM) [22]. Simulation time and memory requirements for microscopic models are high, usually limiting the network size and the number of simulation runs.

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 of the existing network simulators are extended from MANETs before they can be used to simulate vehicular networks.

VANET Simulators: provide both traffic flow simulation and network simulation. Examples are TraNS [23], NCTUns [24], GrooveNet [35], and MobiREAL [36].

III. CRITERIA FOR SIMULATORS

In simulating VANET there are two aspects which need to be considered one is the traffic simulation and other is network simulation. The traffic simulation creates traces of urban mobility model; which is used by network simulation which builds topologies between the nodes and vice versa. There are many problems which exist due non compatibility between traffic simulator and network simulator such as the traces generated once by the traffic simulator seem useless after a certain time as the dynamics of traffic change abruptly. Another problem that remains is the mismatch in formats. The mobility models generated by traffic simulator cannot be processed by the network simulator. The closest network candidate is NS-2, discrete event simulators that accept trace files from other simulators but they can not be fed into NS-2 directly. The criteria for classifying simulators are given below:

Traffic Level Criteria: presents level of details that are concerned with streets, obstruction in communication paths, lights and vehicular densities. For the simulation to capture details at traffic level, it must include the following traces.• **Movement Topologies**: are used to calculate some important factors like speed and distances etc. The topologies are represented with the help of graphs and are classified into the following three types [19]: Custom graphs: Edges are connected by vertex. Random graphs: Using algorithms. Topologies from maps: Graphs from GDF (Geographical Data Files) [17] and TIGER database. **• Start and end position:** The time a node starts its movement marks its initial position is referred to as repelling state, as the node traverses a certain path until it reaches its final position which can be referred to as its attracting point. These two points outline the start and end point for the vehicle. After the graphs are generated, the node's source and destination points are defined for simulation. • **Trip through different positions**: [37] During simulation, the vehicle navigates through different points. These different points are called trip for vehicle •**Selection of track**: The algorithms define the track between paths.• **Speed of vehicles**: The speed of the vehicle depends on the road conditions and can be either smooth or arbitrary.

Motion Level Criteria: After all the details at the traffic level have been captured, the motion level plays its part by creating topologies between the nodes and analyzing their behavior based on the details gathered at traffic level e.g. a car may change its lane and try to overtake. It also monitors the situation during heavy traffic flow or vehicles standing in queue following each other. Motion level feature also defines human behavior patterns through their movement which aids in finding vehicular behavior .Such models are commonly adopted from mathematical equations which produce all possible vehicular behavior patterns. There are various models that fall under this category .The most widely used model is the • **Car following model[40]**: it describes the process of vehicles following each other in the similar line.

Other Criteria: include if the simulator was coupled with a Graphical User Interface (GUI) and also consider the approach to simulate radio obstacles in the wireless communication medium.. There exist many simulators which come under commercial license and some of them are freeware. Various commercial traffic level simulators like AIMSUN [20], VASIM [21], CORSIM [22] etc are powerful commercial traffic simulators aimed at gathering features required for traffic level and with strong GUI support but the traces developed by the traffic simulator cannot be translated for further use by the network simulator and also because of the copy right impendent, these items of software do not fit well with our needs. This paper discusses few simulators which satisfy the above requirements like MOVE[24] , Trans [25], VanetMobiSim [12], NCTUns [22]. Beside the above simulators, there are simulators like CanuMobiSim [18] that has been designed to generate traffic level details. The following simulators generate levels of details at network level like NS [16] , GlomoSim [22]

IV. COMPARISON OF SIMULATORS

A. MOVE (MObility model generator for VEhicular networks)[38] is a Java-based application built on SUMO (Simulation of Urban Mobility) [60] with a facility of GUI.. MOVE can facilitate simulation by generating mobility traces from the TIGER database or Google earth. It also supports custom graphs defined by user and random generated graphs. But with random generated graphs, it restricts the node movement to grid i.e. the node should only move on the grid. MOVE uses parser to extract topological maps from above mentioned tools. MOVE is composed of a Map editor and Vehicular Movement editor. The Map editor creates topological maps for network scenario discussed above. The vehicular movement editor generates movement patterns automatically or can also be defined by the users in the editor. For manual generation, a trip must be defined on the basis of either attraction or repulsion point or randomly. After configuring start and end positions, MOVE can generate random or activity based trip. MOVE calculates path by the mean of Random Waypoint Mobility model or using Dijkstra shortest path first algorithm. Node velocity in MOVE is either smooth or road-dependent. MOVE does not contain any network simulation capabilities but instead parses the traces to be furthered processed by the network simulator. MOVE generates topological maps using parses provided with the map editor and the node parameters that are defined with the help of the vehicular movement editor. This data is then passed to the network simulator. This way they both benefit from interpreters and are able to perform network and traffic tuning. MOVE can also generate its own mobility model but the results obtained are not satisfactory as compared to that of standard mobility models. The problem accompanied with this mobility model is the lack of support for large networks i.e. its packet delivery ratio drops as the number of nodes increase, moreover multiple radio interfaces are not supported by larger networks [1]. While generating mobility traces, MOVE takes micro-mobility into consideration. The micro mobility feature does not include any Lane-changing or Obstacle mobility models. The intersection management follows simplistic stochastic model [2] and therefore random movement of a node in the topology is not considered. The car behavior and interaction with human behavior follows only the car following model. MOVE utilizes the federated approach, in which they both communicate via parser. The traces from the traffic simulators is sent to parser for the translation and then processed by network simulator. The updated file from network simulator is passed to traffic simulator through parser. [40]

B. TraNS (Traffic and Network Simulator environment) [41] is a Java based application with a visualization tool that is built to integrate SUMO and NS-2 specifically designed with VANET simulation in mind. However TraNs has also developed a stepped down version called TraNs Lite for the purpose of generating mobility model only, without using integrated NS-2 simulators for the network simulation. TraNs lite is scalable software with the ability to simulate up to 3,000 nodes and can extract mobility traces from TIGER database or using Shapefile (A vector map, with points, poly lines and polygons) and these maps could be cropped down according to the user's specification. TraCI Traffic Control Interface) interface can combine TraNs lite with ns-2 for traffic and network communication. TraNs utilizes the integrated approach by combining the two well known simulators SUMO and NS-2 inside a single module to facilitate the vehicular simulation. In this way, SUMO translates the traffic file in a form of dump file, which is later on read by a network simulator. The problem with TraNs architecture is that the output obtained from NS-2 cannot be passed back to SUMO, thus the two loosely coupled simulator fails produce the results that are similar to real life examples.

C. VanetMobiSim [41] is an extension to CanuMobiSim and because of its limited scope it can be used in specific areas only as it was unable to produce high levels of details in specific scenarios. Therefore CanuMobiSim was expanded to achieve a high level of realism in the form of VanetMobiSim. Modeling of VanetMobiSim includes car-to-car and car-to-infrastructure relationship. Thus it combines the stop signs, traffic lights and activity based macro-mobility with the support of human mobility dynamics. It can extract road topologies from TIGER, GDF, random and custom topologies. It allows users to generate trips based on their own assumptions or activity based and can configure the path between the start and end position on the basis of the Dijkstra algorithm, road-speed shortest or density-speed shortest. VanetMobiSim contains a parser to extract topologies from GDF, TIGER or cluster Voronoi graphs that will be used by network simulators. The main problem with the above approach is that it does not allow for any feedback among each other. For instance the traces generated by VanetMobiSim are parsed and sent to the network simulator but they cannot feed the data back between each other.[38]

Table I: Traffic Level Features of Various Simulators [38]

Attribute	*SUMO/MOVE/TraNs*	*VanetMobiSim*	*NCTuns*
Custom Graphs	Supports	Supports	Supports
Random Graphs	Grid Based	Voronoi Graphs	SHAPE-File
Graphs from Maps	TIGER database	GDF	Bitmap image
Multilane Graphs	Support	Support	Support
Start/End position	AP, Random	AP, Random	Random
Trip	Random Start - End	Random Start – End	Random
Path	Random Walk, Dijkstra	Random Walk, Dijkstra	Random Walk
Velocity	Road Dependent, Smooth	Road Dependent, Smooth	Road Dependent, Smooth

Table II: Motion Level Features of Various simulators[38]

Attribute	*SUMO/MOVE/TraNs*	*VanetMobiSim*	*NCTuns*
Human Patterns	Car Following Models	Intelligent driver model, Intelligent driver model with intersection management, Intelligent driver model with Lane changes	Intelligent driver model with car following, Intelligent driver model with Lane changing, Intelligent driver model with intersection management
Intersection Management	Stoch turns	Traffic lights and signs	Traffic lights
Lane changing	No Support	MOBIL	Supports
Radio Obstacles	No Support	Supports	Supports

D. NCTUns (National Chiao Tung University Network Simulator*)*[24] is purely written in C++ with a powerful GUI and it can simulate multiple wireless interfaces inside one node including 802.11(p) interface. After the release of version 5, NCTuns enhanced its usability of the ITS project by supporting a large network simulation with the possibility of automatic road assignment using SHAPE-format map file. It uses a car agent module to control vehicular movement dynamic on road with the possibility of autopilot assignments and pre-defined assignment. In autopilot assignment, all the vehicles in the topology are assigned with automatic parameters to control dynamic traffic flow during the simulation.

In a pre-defined assignment, a user has to manually assign values for traffic flow. With its intelligent driving behavior the car agent can model a car to obey certain parameters like traffic light, near by vehicle, changing the lane, taking the turn and car following model. NCTUns implements block objects to introduce the hindering object between wireless signals. The Wall object can completely block the wireless signal or can attenuate the signal with a specified value. The hindering object gives good simulation environment to observe the effects of multi hop wireless network simulation. During the simulation, each node is allowed to send either a UDP or TCP packet. However, there is a limitation in NCTUns. [37] Most of the Network simulators allow multiple TCP/IP versions inside single simulators whereas; NCTUns allows a single instance of TCP/IP version.Unlike TraNs, NCTUns integrates traffic and network simulators inside with a powerful feedback to support vehicular network simulation. However, NCTUns can support a maximum of only 4096 nodes inside a single simulation.

E. NS-2:[41] code is written either in C++ and OT CL and is kept in a separate file that is executed by OTCL interpreter, thus generating an output file for NAM (Network animator). It is packaged with a bundle of rich libraries for simulating wireless networks. For simulating a wireless node the physical layer, the link layer and MAC (media access control) protocol are all included at the same time. But despite this NS-2 is unable to simulate multiple radio interfaces. Moreover NS-2 has unrealistic models for wireless channel, which results in a biased radio propagation. For wireless simulation, NS-2 supports only free space and two ray ground reflection models and cannot simulate path loss, multi-path fading and shadowing phenomena. NS-2 has certain limitations when it comes to including more than one wireless interfaces per node. NS-2 only supports Bi-directional (antenna that radiates or receives most of its energy in two directions) and Omni-directional (radiates signal equally in all direction) antenna for signal propagation and waypoint mobility model for node movement. The newly added modules cannot find their place in the outdated documentation. Also simulating large networks takes a toll over the CPU cycles and memory management. NS-3 came around as a better replacement for its predecessor. [41] NS3 is written purely in C++ and limits the coding to only a few hundred lines as opposed to 300,000 lines for that of NS-2. For simulating huge networks NS3 was equipped with support for distributed and federated simulation tasks. Work is still underway on NS3 and it is hoped that it will completely take over its predecessor sooner or later.[38]

F. GlomoSim (*Glo*bal *Mo*bile Information System *Sim*ulator)[19] is a second most popular network simulator after NS-2. GlomoSim has the ability to run on SMP (shared-memory symmetric processor: memory simultaneously accessible by all programs) and helps to divide the network into separate modules each running as a distinct process. Because of this extraordinary feature of multi-tasking, GlomoSim is able to simulate tens of thousands of nodes in single simulation. For most of the simulators, the focal point is simulating Random Waypoint mobility model.[41] To enhance the functionality, the GEM project was added to GlomoSim to have more realistic simulation. GlomoSim follows OSI layer model and support different protocols and models at each layer. Unlike NS-2, GlomoSim has the ability to support multiple wireless technologies including IEEE802.11e.]GlomoSim has Two-ray and free-space radio propagation models. SNR (Signal to noise interference ratio) is cumulative i.e. each time receiver predicts the change in interference power and recalculates SNR of the given signals.[38] GloMoSim was designed to support millions of nodes just as a single simulation is due to parallelism technique.

Table III: Motion Level Features of Network Simulators [38]

Simulator	*GloMoSim*	*NS-2*	*NCTuns*	
Signal to Noise Ratio Calculation	Cumulative	Difference in two Signals	Cumulative	
Signal Reception	SNRT, BER	SNRT	Sender	Receiver
			Transmitting power	Power threshold, Distance
Fading	Rayleigh, Ricean	No	Rayleigh, Ricean	
Path Loss	Free Space, Two Ray	Free Space, Two Ray	Free Space, Two ray, Free space with shadowing	
Support for Multiple Wireless Technology	Yes	No	Yes	
Antenna's Support	Bi-directional, Omni-directional	Bi-directional, Omni-directional	Directional, Bi-directional, Rotating	
Distributed Simulation	Yes	No	Yes	
Time required for Simulating 5000 Nodes (sec)	6191	Fail	Fail	
Memory Required for Simulating 5000 Nodes (KB)	27.5	Fail	Fail	
GUI	Yes	No	Yes	

V. CONCLUSION & FUTURE WORK

This paper discusses broadly all types simulators used in VANET as it is easier to check all the constraints in a simulated environment rather than real scenario since outdoor experiments are costly and may not provide us with all the necessary stimuli.This requires the use of realistic mobility models generating realistic mobility patterns. VANET simulation requires that a traffic and network simulator can be jointly used with feedback between them to render the simulation results as accurate as real life. This paper covers important traffic and network with their features. The problem faced by

VanetMobiSim and NS-2. [28, 29] are that traces are generated only once and no feedback is allowed. The problem faced by TraNs [36] is Loose coupling and the feedback is slow. The problem faced by MOVE and NS-2 / QualNet is lack of interaction.[38] The review presented in this paper can be extended further by searching or developing for simulators which can run with both real and modeled vehicles and capture the traffic directly from a video camera and can also identify radio obstacles like the effect of rain, fog, magnetic field.

REFERENCES

[1] A survey and comparative study of simulators for vehicular *ad hoc* networks (VANETs) Francisco J. Martinez, Chai Keong Toh, Juan-Carlos Cano, Carlos T. Calafate and Pietro Manzoni

[2] Toh C-K. *Ad Hoc Mobile Wireless Networks: Protocols andSystems*. Prentice Hall: Upper Saddle River, NJ, USA, 2001.

[3] Cavin D, Sasson Y, Schiper A. On the accuracy ofMANET simulators.In *Proceedings of the 2nd ACM International Workshopon Principles of Mobile Computing*.ACM:NewYork,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.paramicsonline.com/

[7] Capra L, Emmerich W, Mascolo C. Carisma: context-aware reflective middleware system for mobile applications. *IEEE Transactions on Software Engineering* 2003

[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.scalablenetworks.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 VehicularNetworks 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.

[16] Haerri J, Fiore M, Fethi F, Bonnet C. VanetMobiSim:generating realistic mobility patterns for VANETs. Institute Eurécom and Politecnico Di Torino, 2006. Available at:http://vanet.eurecom.fr/

[17] 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/nsdocumentation. html

[18] Martin J. GloMoSim. Global mobile information systems simulation library. UCLA Parallel Computing Laboratory, 2001. Available at: http://pcl.cs.ucla.edu/projects/glomosim/

[19] 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/

[20] JiST/SWANS: Java in Simulation Time/Scalable Wireless Ad hoc Network Simulator, 2004. Available at: http://jist.ece. cornell.edu/

[21] The Georgia Tech Network Simulator (GTNetS), 2008. Available at: http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/

[22] Piorkowski M, Raya M, Lugo AL, Papadimitratos P, GrossglauserM, Hubaux J-P. TraNS (Traffic and Network Simulation Environment). Ecole Polytechnique Fédérale de Lausanne, EPFL, Switzerland, 2007. Available at: http://trans.epfl.ch/

[23] NCTUns 5.0, 2008. Available at: http://nsl10.csie.nctu.edu.tw/

[24] 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/

[25] MobiREAL, 2008. Available at: http://www.mobireal.net/

[26] Choffnes DR, Bustamante FE. Modelling vehicular traffic andmobility for vehicular wireless networks. Technichal Report, Department of Computer Science, Northwestern University,NWU-CS-05-03, July 2005.

[27] The network simulator ns-2, http://www.isi.edu/nsnam/ns

[28] VanetMobiSim Project Home Page, http://vanet.eurecom.fr

[29] Fiore, M., Harri, J.; Filali, F. and Bonnet, C. (2008), "Vehicular Mobility Simulation for VANETs Simulation" Symposium, ANSS apos; 07. 40th Annual Volume, Issue , 26-28, pp: 301 – 309

[30] Fiore,M. (2006), "Mobility Models in Inter-Vehicle Communications Literature", Technical Report.

[31] Agüero, R and Pérez, J. (2007), "Adding Multiple Interface Support in NS-2", Available at http://personales.unican.es/aguerocr

[32] Romano, N. C. and Numamaker, J. F. (2001), "Meeting analysis: Findings from research and practice," Proceedings of Teh 34th Hawaii International Conference on Systems Science

[33] Liu, B., Khorashadi, B., Du, H.N., Ghosal, D., Chuah. C.N., and Zhang,M. (2009) "VGSim: An integrated networking and microscopic vehicular mobility simulation platform", IEEE Press, pp: 134-141

[34] Karnadi, F.K.; Mo, Z.H.; Lan,K.C. (2007), "Rapid Generation of Realistic Simulation for VANET", School of Comput. Sci. & Eng., New South Wales Univ., Sydney, NSW, Wireless Communications and Networking Conference, WCNC 2007. IEEE.

[35] Piorkowski, M., Raya, M., Lugo, A., Papadimitratos, P., Grossglauser, M., Hubaux, J.P. (2008) "TraNS: Realistic Joint Traffic and Network Simulator for VANETs", ACM SIGMOBILE Mobile Computing and Communications Review, vol. 12, num. 1

[36] Wang, S.Y. and Chou, C.L. (2009) "NCTUns Tool for Wireless Vehicular Communication Network Researches," Simulation Modelling Practice and Theory, Vol. 17, No. 7, pp. 1211-1226.

[37] Aamir Hassan, Tony Larsson, "On the requirements on models and simulator design for integrated VANET Simulation" 8th International Workshop on Intelligent, 2011.

[38] Kubach, U., Becker, C., Stepanov, I. and Tian, J., (2004) "Simulation Model and Tool for Mobile Location Dependent Information Access". In "Mobile Computing Handbook", Eds. M. Ilyas, I. Mahgoub, CRC Press, New York.

[39] Ruben,M., Jean-Yves,L.B. and Jörg,W. (2007), " An Architecture for Wireless Simulation in NS-2 Applied to Impulse-Radio Ultra-Wide Band", Networks 10th Communications and Networking Simulation Symposium; Norfolk, VA, March 25-29

[40] Aamir Hassan "VANET Simulation", School of Information Science, Computer and Electrical Engineering, Halmstad University 2009

