UNIVERSITY OF VAASA

FACULTY OF TECHNOLOGY

TELECOMMUNICATION ENGINEERING

Hayder Mohammed Ali

SIMULATION AND ANALYSIS OF VEHICULAR AD-HOC NETWORKS IN URBAN AND RURAL AREAS

Master´s thesis for the degree of Master of Science in Technology submitted for inspection, Vaasa, 20th of December, 2013.

Supervisor Mohammed Elmusrati

Instructor Tobias Glocker

ACKNOWLEDGMENTS

First of all, praise be to Allah my light in the darkness and hope in adversities for supporting me my entire life.

Then I would like to express my gratitude towards my supervisor Mohammed Elmusrati for his guidance and to give special thanks for my instructor Tobias Glocker for his continuous and invaluable support and help throughout the whole thesis work period.

I would like to extend my thanks to my teachers Mulugeta Fikadu, Tommi Sottinen, Matti Linna and Reino Virrankoski for their great courses and guidance throughout my studies.

I'm grateful for the University of Vaasa and the people and government of Finland for giving me the opportunity to study for free in their great program of telecommunication engineering and for providing all the necessary material and equipment to succeed in my studies.

I want also to thank the engineers at the telecommunication group in the University of Vaasa, my classmates and my dear friends who made my last two year an unforgettable pleasant experience.

Finally, I would like to greatly thank my family and my dearest mother and father for raising me to be the man I am today and I feel proud to present this work for my departed farther may Allah have mercy on his soul.

Hayder Mohammed Ali

Vaasa, Finland, 5 December 2013

TABLE OF CONTENTS **PAGE**

LIST OF FIGURES **PAGE**

LIST OF TABLES **PAGE**

ABBREVIATIONS

A-STAR	**Anchor-Based Street and Traffic Aware Routing**
ABR	**Associativity-Based Routing**
ACK	**Acknowledgment**
AGF	**Advanced Greedy Forwarding**
AMAR	**Adaptive Movement Aware Routing**
AODV	**Ad Hoc on Demand Distance Vector**
ARP	**Address Resolution Protocol**
B-MFR	**Border-Node Based Most Forward within Radius**
CAR	**Connectivity Aware Routing Protocol**
CBDRP	**Cluster-Based Directional Routing Protocol**
CBR	**Constant Bit Rate**
CFM	**Car Following Models**
CGSR	**Cluster-head Gateway Switch Routing**
CTS	**Clear to Send frame**
DAG	**Directed Acyclic Graph**
DIR	**Diagonal-Intersection-based Routing Protocol**
DRG	**Distributed Robust Geocast**
DSDV	**Destination-Sequenced Distance-Vector Routing**
DSR	**Dynamic Source Routing**
DSRC	**Dedicated Short-Range Communication**
DTN	**Delay Tolerant Network**
DTSG	**Dynamic Time-Stable Geocast Routing**
EBGR	**Edge Node Based Greedy Routing Protocol**
EDR	**Event Data Recorder**
FIFO	**First in First Out**
FSR	**Fisheye State Routing**
GDF	**Geographical Data File**
GPCR	**Greedy Perimeter Coordinator Routing**
GPS	**Global Positioning System**

GPSR	**Greedy Perimeter Stateless Routing**
GRANT	**Greedy Routing with Abstract Neighbor**
GSR	**Geographic Source Routing**
GUI	**Graphical User Interface**
GyTAR	**Greedy Traffic Aware Routing Protocol**
HARP	**Hybrid Ad Hoc Routing Protocol**
IARP	**Intra-Zone Routing Protocol**
IDM	**Intelligent Driver Model**
IEEE	**Institute of Electrical and Electronics Engineers**
IERP	**Inter-Zone Routing Protocol**
IP	**Internet Protocol**
IPv6	**Internet Protocol version 6**
ISI	**Inter Symbol Interference**
IVG	**Inter-Vehicle Geocast**
LORA-CBF	**Location Routing Algorithm with Cluster-Based Flooding**
LOS	**Line of Sight**
MANET	**Mobile Ad-Hoc Network**
MOVE	**Mobility Model Generator for Vehicular Networks**
MPR	**Multipoint Relays**
NCTUns	**National Chiao Tung University Network Simulator**
ns-2	**network simulator 2**
OFDM	**Orthogonal Frequency Division Multiplexing**
OLSR	**Optimized Link State Routing Protocol**
OSR	**Open Street Map**
pdf	**probability density function**
PGB	**Preferred Group Broadcasting**
ROMSGP	**Receive on Most Stable Group-Path**
ROVER	**Robust Vehicular Routing**
RPGM	**Point Group Mobility Model**
RSU	**Rode Side Unit**
RTS	**Request to Send frame**
RWP	**Random Way Point**

TBRPF	**Topology Dissemination Based on Reverse-Path Forwarding**
TCP	**Transmission Control Protocol**
TIGER	**Topologically Integrated Geographic Encoding and Referencing**
TORA	**Temporally Ordered Routing Algorithm**
TraNS	**Traffic and Network Simulation Environment**
STBR	**Street Topology Based Routing**
STRAW	**Street Random Waypoint**
SUMO	**Simulation of Urban Mobility**
UDP	**User Datagram Protocol**
UMB	**Urban Multi-hop Broadcast Protocol**
UMTS	**Universal Mobile Telecommunications System**
V-TRADE	**Vector Based Tracing Detection**
V2I	**Vehicle to Infrastructure**
V2V	**Vehicle to Vehicle**
VADD	**Vehicle-Assisted Data Delivery Routing Protocol**
VANET	**Vehicular Ad Hoc Network**
VGPR	**Vertex-Based predictive Greedy Routing**
WAVE	**Wireless Access for Vehicular Environment**
WLAN	**Wireless Local Area Network**
WRP	**Wireless Routing Protocol**
XML	**Extensible Markup Language**
ZOF	**Zone of Forwarding**
ZOR	**Zone of Relevance**
ZRP	**Zone Routing Protocol**

UNIVERSITY OF VAASA
Faculty of Faculty of technology

Author:	Hayder Mohammed Ali
Topic of the Thesis:	Simulation and Analysis of Vehicular Ad-hoc Networks in urban and rural areas
Supervisor:	Mohammed Elmusrati
Instructor:	Tobias Glocker
Degree:	Master of Science in Technology
Department:	Department of Computer Science
Degree Programme:	Degree Programme in Information Technology
Major of Subject:	Telecommunications Engineering
Year of Entering the University:	2011
Year of Completing the Thesis:	2013 **Pages:** 131

ABSTRACT

According to the American National Highway Traffic Safety Administration, in 2010, there were an estimated 5,419,000 police-reported traffic crashes, in which 32,885 people were killed and 2,239,000 people were injured in the US alone. Vehicular Ad-Hoc Network (VANET) is an emerging technology which promises to decrease car accidents by providing several safety related services such as blind spot, forward collision and sudden braking ahead warnings. Unfortunately, research of VANET is hindered by the extremely high cost and complexity of field testing. Hence it becomes important to simulate VANET protocols and applications thoroughly before attempting to implement them. This thesis studies the feasibility of common mobility and wireless channel models in VANET simulation and provides a general overview of the currently available VANET simulators and their features. Six different simulation scenarios are performed to evaluate the performance of AODV, DSDV, DSR and OLSR Ad-Hoc routing protocols with UDP and TCP packets. Simulation results indicate that reactive protocols are more robust and suitable for the highly dynamic VANET networks. Furthermore, TCP is found to be more suitable for VANET safety applications due to the high delay and packet drop of UDP packets.

KEYWORDS: vehicular ad-hoc networks, routing protocols, TCP, UDP, mobility models, wireless channel models, simulation,

1. INTRODUCTION

Vehicular Ad Hoc Networks (VANETs) are a subgroup of mobile ad hoc networks (MANETs) with the distinguishing property that the nodes are vehicles like cars, trucks, buses and motorcycles. These nodes are highly mobile and they are able to communicate with each other by Vehicle to Vehicle (V2V) communications and to connect to the infrastructures by Vehicle to Infrastructure (V2I) communications for services. The main goals of implementing VANETs are safety, efficiency and environmental friendliness. Between car makers and highway agencies, the main problem has always been funding; who will pay? And is it worth to invest?

In 1970, an Electronic Route-Guidance System was proposed in USA. The driver provides the vehicle with a code word for destination. At every intersection the car sends the code word to the roadside which replies with instructions to the driver. In Japan, Comprehensive Automobile Traffic Control System was carried from 1973 to 1979 to reduce traffic congestion and exhaust fumes and to prevent accidents. In 1986, PROMETHEUS program was initiated in Europe and it includes three sub-programs for driving assistance, V2V and V2I communications. Starting from 1990 and forward, several research activities were carried on in Japan, USA and Europe focusing on *cooperative driving assistance* and safety applications. In 1999, the US Federal Communication Commission allocated 75 MHz bandwidth of the 5.9 GHz band to the Dedicated Short-Range Communication (DSRC). In 2004, IEEE started to work on the 802.11p and the WAVE standards to be part of the DSRC. In 2005, the US department of transportation demonstrated practically several applications for VANETs such as electronic payments, vehicles location and speed data collection and displaying messages and traffic signal indications for the driver. Research results showed that IPv6 performed well in VANETs, demonstrated the lack of accuracy in GPS receivers and proved the existence of a strong relation between the availability of a Line of Sight (LOS) communication and achieving low packet error rates. (Hartenstein & Laberteaux 2010: 4-7.)

DSRC is undergoing intensive research and development focusing on the area of collision prevention applications. (Kenney 2011: 1162.)

Following are some of the common characteristics and challenges facing VANETs.

Complicated characteristics of radio channel: The presence of buildings, road signs, trees, and obstacles with different shapes, dimensions and materials makes it rather difficult to model the radio channel. Furthermore, the nodes themselves (the cars) contribute to the signal reflection, refraction and scattering. They are continuously shifting in position and changing in density which increases the dynamic nature of the radio channel making it harder to predict. (Hartenstein & Laberteaux 2010.)

Lack of centralized management: Without central management, there is no synchronized transmission between different nodes which might result in a high packet collision rate thus reducing transmission efficiency. (Hartenstein & Laberteaux 2010.)

High speed of vehicles: This causes fast paced changes in the network topology. Disconnection can happen in a matter of seconds when one car exits the transmission range of another due to the difference in moving velocity. Nodes continuously and rapidly enter or exit the network triggering fast unpredictable changes in the availability of routes between the source and destination nodes. As a result, routing packets between two nodes becomes much more complicated task compared to the traditional MANETs and it becomes necessary to design and implement more sophisticated routing protocols tailored specifically to the needs of VANETs applications. (Paul, Ibrahim & Bikas 2011.)

Mobility patterns differ greatly between MANETs and VANETs: In MANETs nodes have free and random movement patterns. On the other hand, VANET nodes movement is restricted by the topology of the streets, traffic signs and speed limits. This requires different approaches when modeling node’s movement. (Paul, Ibrahim & Bikas 2011.)

Security: Security vulnerabilities and threats are inherited in all kinds of wireless networks and VANETs are not an exception.

Standardization is necessary: Because there are various car and equipment manufacturers each with different hardware and technologies when it comes to VANETs applications. However, strong competition in the market requires each manufacturer to have their own unique services to differentiate themselves from the competitors. (Hartenstein & Laberteaux 2010.)

VANET implementation: Concerns road infrastructure, public transportations, car manufacturers and individuals owning the vehicles. Implementation and testing is too expensive and complicated. As a result there is a lack of practical cost-benefit analysis and most research efforts use simulators. (Hartenstein & Laberteaux 2010.)

Most modern cars are assumed to have unlimited power sources: The size of the car makes it possible to install powerful and heavy equipment which is not possible in most cases of MANETs (Yousefi, Mousavi & Fathy 2006).

The thesis consists of seven chapters. The first chapter introduces the topic of VANETs, their history, main characteristics and challenges. Chapters 2 and 3 present some of the common mathematical mobility and wireless channel models which can be used in VANET simulations. Chapter 4 illustrates the most commonly discussed routing protocols in VANET’s scientific research papers. Chapter 5 surveys some of the currently available VANET simulators and demonstrates their main features and shortcomings. Chapter 6 explains the software and methods used for simulations in this thesis. Each simulation scenario is explained briefly. Results are presented in a form of graphs, tables and charts along with their analysis and discussions. Chapter 7 summarizes key finding obtained from simulations, gives recommendations and suggests future work plans.

2. MOBILITY MODELS.

Mobility models describe the movement patterns of nodes communicating in a Vehicular Ad-Hoc Network. They play a vital role in simulation and have a great impact on performance evaluation of routing protocols. The functions of routing protocols such as discovering, maintaining or reconstructing routes are highly influenced by the movement patterns of the nodes. Changes in topology and node movement speed affect the decision when choosing which node is the best to be utilized as a relay for forwarding packets. The Random Way Point (RWP) mobility model was used widely in earlier VANET simulations. It is originally a MANET mobility model which assumes free random movement of nodes inside the simulation area without considering any obstacles. However, in VANET, the movement of nodes is restricted by the street layout, traffic lights, speed limits and obstacles. Furthermore, it is governed by various factors such as car acceleration and deceleration, traffic congestion and jams, queuing at intersections, weather condition and even the mood and feelings of the human driver. Lot of research has been carried out to develop more realistic mobility models for VANET simulations. (Khairnar & Pradhan 2011.)

2.1. Factors affecting mobility in VANETs

Street Layouts: Streets restrict nodes movements within pre-defined paths instead of random ones. Streets have parameters such as width and physical condition and attributes such as being single or multilane, the possibility of overtaking and whether the street is one or two ways. All of these factors must be taken into consideration when building a mobility model for VANET simulation. (Mahajan, Potnis, Gopalan & Wang 2006.)

Block size: A city block can be considered as the smallest area surrounded by streets. A smaller block size means more intersections forcing vehicles to stop more frequently. A larger block size would increase the effect of clustering. (Mahajan et al. 2006.)

Traffic control mechanisms: Stop signs and traffic lights at intersections force vehicles to stop or reduce their speed resulting in queues and clusters of vehicles. Slower average speed of nodes results in more stable topologies and thus better throughput. On the other hand, clustering has a negative impact and degrades the network performance. (Mahajan et al. 2006.)

Influence of surrounding nodes: Drivers tend to keep minimum distance between their car and the one in front of it. This also implies a dependency in movement speed, acceleration and deceleration between two successive nodes. The possibility of switching to another lane to overtake the preceding car should also be taken into consideration. (Mahajan et al. 2006.)

Movement Speed: Determines the frequency of changes in topology and the rate of link breakdown. Routing protocols performance is directly influenced by the average speed of nodes movement which is dependent on the speed limit of a given area. Also street characteristics such as the numbers of intersections, length of straight sections and block size affect the acceleration and deceleration of cars which in turn will have an impact on network performance. (Mahajan et al. 2006.)

Weather Conditions: Snow, rain and fog can decrease driver's visibility and control over car thus affecting movement, acceleration and declaration speed. In some cases, weather conditions can physically obstruct cars movement or even change the street layout.

Node Density: Depends on both place (small town or large crowded city) and time (rush hour or late at night)**.** A higher density results in better availability of intermediate relay nodes. On the other hand, increasing the number of hops will induce more delay and increase the packet overhead size thus decreasing throughput. Depending on their architect, different routing protocols will perform differently when varying cars density.

Influence of Time: Cars density varies greatly during one day. Rush hours occur in the morning and evening but in different segments of the roads (going to or back from work) while streets become almost empty in some cities after midnight. Furthermore, visibility and physical statues of the driver differ greatly between day and night thus their driving behaviors will also differ.

2.2. Classification of Mobility Models

2.2.1. Random Mobility models

Vehicular mobility parameters such as speed, direction and destination are sampled randomly from stochastic processes. They are easy to implement and to analyze. However, they offer very limited interactions between nodes and that does not reflect the reality of VANETs. For example, in the RWP model, each vehicle samples its next destination and speed randomly and moves in a fixed speed. In the Reference Point Group Mobility Model (RPGM), nodes are split into groups which follow their leader’s speed and direction of motion with a small probability of deviation. More realistic random mobility models include the freeway model which restricts nodes movement in bi-directional multi-lane freeways and the Manhattan model which restricts movement to urban grids. (Hartenstein & Laberteaux 2010: 113.)

2.2.2. Flow Models

They take into consideration interactions among vehicles and between them and the surrounding environment. Flow models are classified into microscopic, mesoscopic and macroscopic models depending on the level of details provided for the interaction between vehicles. These models can be either single or multi-lane models which allow cars overtaking. In that case, three parts must be evaluated; driver's need for lane changing, the feasibility of lane changing and its trajectory. More realistic flow models would take into account traffic signs and speed limits as well. (Hartenstein & Laberteaux 2010: 115.)

2.2.2.1. Microscopic Models

These models are computationally complex. They provide precise modeling for every car's mobility patterns such as acceleration/deceleration, minimum distance to the preceding car and even driver's behavior and reaction time are simulated based on a behavioral theory. They are mainly developed to simulate accident free environments. (Hartenstein & Laberteaux 2010: 116.)

Figure 1. Illustrates some of the microscopic mobility models.

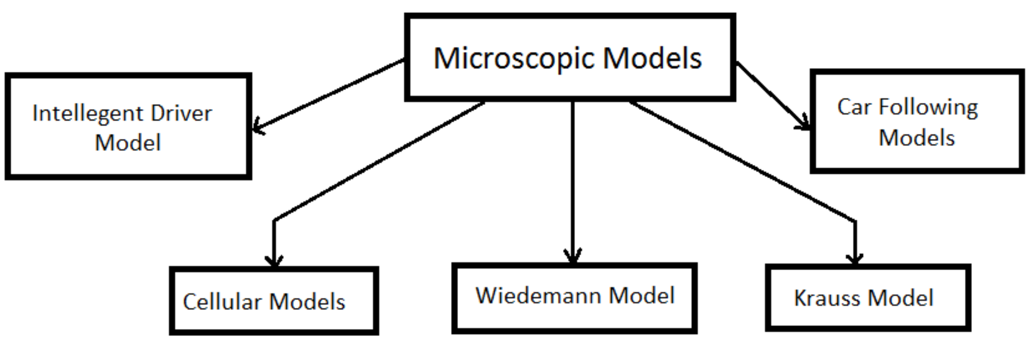


Figure 1. Microscopic Mobility Models.

In Car Following Models (CFM), each individual car maintains a safe inter-distance to the leading car in order to avoid collision. Distance between vehicles, speed and acceleration are represented as continuous functions. Pipe’s rule describes the safety distance between two vehicles as the minimum distance for a driver to completely stop without hitting the leading vehicle.

The safety distance is given in Pipe’s rule as a function of the car’s speed:

$$\Delta x^{safe}(v_i) = L + T.v_i + \psi.v_i^2 \qquad (1)$$

Where *L* is the vehicle length, *T* is the reaction time, $\psi.v_i^2$ is the breaking distance, and ψ is for adjusting deceleration. When the breaking distance between the two vehicles is equal, ψ is set to 0. If the leading vehicle suddenly stops, ψ is maximized. (Pipe, 1953; Hartenstein & Laberteaux, 2010: 117.)

Acceleration is modeled as a function of perceived relative speed between the leading and following car. An example is the Gazis–Hermann–Rothery (GHR) model also known as the General Motor (GM) model which describes acceleration of a vehicle at time *t* in terms of its speed and distance difference to the leading vehicle at time (t – T).

$$\frac{dv_i}{dt}(t) = c\,.\,v_i^m\,(t).\frac{\Delta v_i\,(t-T)}{\Delta x_i^l\,(t-T)} \qquad (2)$$

where v_i is the speed of the following vehicle, *T* is the following vehicle’s driver reaction time, *c* is an adjusting coefficient, *m* is the speed exponent with values between -2 to 2 and *l* is a distance exponent with values in the range of -4 to 1. (Hartenstein & Laberteaux, 2010: 118.)

The Intelligent Driver Model (IDM) is also based on driver's stimulated response. Instead of the perceived speed difference, it considers the stimulus to be the driver's desired distance gap to the leading vehicle. This can be expressed in equation 3.

$$\frac{dv_i}{dt}(t) = a\left(1 - \left(\frac{v_i(t)}{v_i^{des}}\right)^4 - \left(\frac{\delta(v_i(t), \Delta v_i(t))}{\Delta x_i(t)}\right)^2\right) \tag{3}$$

Where v_i^{des} is the desired speed, Δx_i is the current gap and δ is the desired gap. (Hartenstein & Laberteaux, 2010: 118.)

The desired gap is calculated in equation 4.

$$\delta(v_i(t), \Delta v_i(t)) = \Delta x^{rest} + \left(v_i(t).\mathrm{T} + \frac{v_i(t).\Delta v_i(t)}{2\sqrt{a.b}}\right) \tag{4}$$

Where Δx^{rest} is the gap between two vehicles at rest, *a* is the maximum acceleration and *b* is the maximum deceleration. (Hartenstein & Laberteaux 2010: 118.)

The Krauss model is a discrete time model which computes the future speed at time intervals as a stochastic process using maximum speed, acceleration and deceleration as input parameters. (Hartenstein & Laberteaux 2010.)

The Wiedemann model considers driver's mentality and generates different responses for same stimulus. The driver is considered to be in one of the following four driving modes:

- When there is no leading vehicle nearby, the driver freely accelerates until he reaches his desired speed.
- Approaching mode: Decelerating until he reaches a safe inner-distance to the leading vehicle.
- Following Mode: Same as CFM with smooth acceleration and deceleration.

- Breaking Mode: Tries to avoid crashing into the leading vehicle by applying high deceleration.

Cellular Models are discrete in both time and space which reduces computational complexity. A lane is represented as a frame of equally sized cells. Cars navigate between those cells with their speed expressed as the number of cells per time step. (Hartenstein & Laberteaux 2010.)

2.2.2.2. Macroscopic Models

These models focus on overall dynamic flow of large groups of vehicles. They describe the speed, density and flow of vehicles at a defined location and time. Generally, macroscopic models are represented by three equations in three unknowns. The first one describes the relation between flow (m), velocity (v) and density (ρ) (Hartenstein & Laberteaux, 2010: 121.)

$$m(x, t) = \rho(x, t) \cdot v(x, t) \tag{5}$$

The second states that the density of cars as a rate of time in a position x varies according to the flow of cars into or out of that position. (Hartenstein & Laberteaux, 2010: 121.)

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial m(x,t)}{\partial x} = 0 \tag{6}$$

The final equation is model dependent. For example, the most know macroscopic model Lighthill–Whitham–Richard (LWR) describes velocity in terms of density leading to a third equation. (Hartenstein & Laberteaux, 2010: 121.)

$$\frac{\partial\rho(x,t)}{\partial t} + \frac{\partial\rho(x,t)\cdot v(\rho(x,t))}{\partial\rho} \cdot \frac{\partial\rho(x,t)}{\partial x} = 0. \tag{7}$$

Macroscopic models offer simplified computational complexity at the expense of less precision compared to microscopic models. They are well suitable for modeling large-scale traffic scenarios such as a whole city. Unfortunately, they fail to model clustering effect which is crucial for urban VANET simulations.

2.2.2.3. Mesoscopic Models

Traffic flow is modeled as a probability density function while interactions between individual vehicles and clustering effects are also taken into consideration. A typical example is the queue model in which road segments are presented by First in First out (FIFO) queues with macroscopic characteristics governing individual cars behaviors. Each queue has a limited capacity as well as a maximum ongoing capacity. Therefore, a car cannot switch queues unless its own queue does not exceed its ongoing capacity and the target queue still has a free place to accept it. (Hartenstein & Laberteaux 2010: 123.)

2.2.3. Traffic Models

Traffic Models are responsible for modeling the path followed by a car (Origin-Destination or OD), turning behavior (predetermined or stochastic) as well as traffic lights and stop signs at intersections. They consist of two parts; trip planning and path planning where both are influenced by time and human behavior. The origin and destination of a trip planning is decided based on the person's residence place and needs (going to work shopping, etc.). The selected path is also not random; the driver will usually select the fastest or least crowded path based on his or her personal experience and driving habits. (Hartenstein & Laberteaux 2010: 131.)

Trip Planning: There are three common techniques used to model trip planning. The simplest is to let cars select their origin and destination points randomly without any correlation between them. In stochastic turn technique, a path is not planned. At every intersection a new direction is determined by stochastic equations based on field measurements. Alternatively, field surveys are conducted to identify important points on a map (such as landmarks) and their turning probabilities. The data is then used to build origin-destination matrices which take into account the correlation between various trips. (Hartenstein & Laberteaux 2010: 132.)

Path Planning: Path planning determines the sequence of directions followed by cars to travel from the origin to the destination point based on a preferred optimization function such as shortest, fastest or least crowded path. These parameters are dynamic over time and are also influenced by the chosen path itself. Therefore, paths need to be recalculated periodically and alternative paths need to be pre-computed and to be integrated into driver's choices. As a result, path planning is a challenging and resource intensive task. (Hartenstein & Laberteaux 2010.)

2.2.4. Behavioral Models

Human driving behaviors are far too complex to be modeled by specific synthetic models. They are influenced by the driver's physical condition, social habits (country specific), different perception toward traffic scenarios, road condition and obstacles, time and weather. Behavioral models try to implement artificial intelligence to mimic unique reactions for every individual driver. However, they fall short due to their tremendous computational complexity. An alternative approach is to develop a set of unique driving habits and strategies called Driver Agents then a percentage from total cars is associated with every driving agent. (Hartenstein & Laberteaux 2010: 133.)

2.2.5. Survey Based Models

Instead of building complex synthetic models and later modify them to match real world scenarios, it is wiser to survey cars mobility patterns in cities and use their traces to build realistic mobility models directly . This approach however, is currently hindered by the limited availability of vehicular traces around the cities of the world. (Hartenstein & Laberteaux 2010: 135.)

3. WIRELESS CHANNELS

Cars always move on roads. However, those roads can exist on a variety of geographical scenes. They can be in completely open spaces such as deserts, farms or aside the beach. They can be placed on top of mountains with great variance in surface altitude or inside forests where there are lots of trees. Cities can be crowded and full of large buildings and sky scrapers like New York or spacious and less crowded like Vaasa. Consequently, signals propagating in a wireless medium between VANET nodes will be affected differently depending on the unique characteristics of the landscape. These effects are categorized into small scale effects and large scale effects.

3.1. Large Scale Effects

Noticeable when signals travel long distances between the transmitter and the receiver thus called large scale. Signal's power will be reduced due to the path loss which can be expressed approximately by equation 8.

$$\Gamma_{dB} = 10v\,.\log\left(\frac{d}{d_0}\right) + c \tag{8}$$

where Γ_{dB} is the path loss measured in dB, d is the distance between the transmitter and the receiver, ν is the path loss exponent, c is a constant, and d_0 is the distance to power measurement reference point. Typical values for v would range 2 (free space) to 6. The constant c is derived from the physical attributes of the transmitter such as signal wavelength and antenna height. (Liu, Sadek, Su & Kwasinski 2009: 4.)

In reality, path losses do not only depend on the distance between transmitter and receiver but on which objects are actually obstructing wave's propagation. These obstructions are unknown beforehand. Thus their effects can be modeled as a random variable. A more practical representation for the large scale fading is shown in equation 9 and it is called shadow fading.

$$\Gamma_{dB} = 10v \, . \log\left(\frac{d}{d_0}\right) + S + c \qquad (9)$$

Where S is the random variable which accounts for the shadow loss.
"It has been found through experimental measurements that S when measured in dB can be characterized as a zero-mean Gaussian distributed random variable with standard deviation σ (also measured in dB)." (Liu, Sadek, Su & Kwasinski 2009: 5.)

3.2. Small Scale Effects

Small scale effects are noticeable over short distances in the order of the transmitted signal's wavelength. They result from the presence of obstacles such as buildings, cars, signs and trees between the transmitter and receiver antennae. When encountering an obstacle, an electromagnetic wave will be reflected, diffracted or scattered depending on the nature and dimensions of the surface of that obstacle.

Reflection: occurs when the surface is smooth and very large compared to the signal's wavelength. (Heikki & Elmusrati 2009: 70.)

Refraction: If the obstructing object is a dense large body, points on the wave front act as the seed causing secondary waves to form behind the obstacle and to reach the receiver's antenna even though it is being shadowed by an impenetrable obstructing body. (Heikki & Elmusrati 2009: 70-71.)

Scattering: In case the encountered surface is a rough body with dimensions equal to or smaller than the wavelength (for example lamp posts and street signs), the signal will be scattered in all directions. (Heikki & Elmusrati 2009: 71.)

As a result, rapid fluctuations occur in the received signal's amplitude and phase caused by the arrival of multiple different versions of the original transmitted signal. Those versions are called multipath signals because they arrive with different delays and from different directions. They combine together at the receiver's side causing the received signal to distort or fade. Assuming that the channel is linear and does not change over time, the relation between the transmitted signal *x(t)* and the received signal *y(t)* can be written as:

$$y(t) = \sum_{i=0}^{L} h_i \; x(t - \tau_i) \tag{10}$$

Where h_i is the attenuation of the i-th path and τ_i is its corresponding time delay. (Liu, Sadek, Su & Kwasinski 2009: 6.)

Figure 2 illustrates the phenomena of scattering reflecting and refraction.

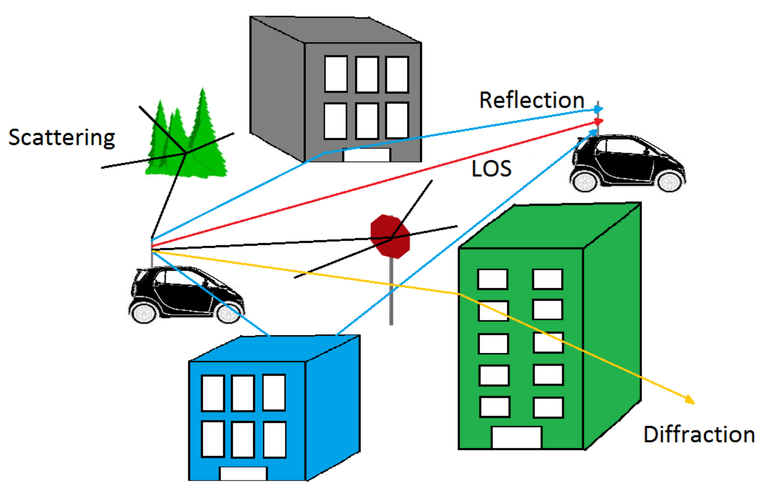


Figure 2. Reflection, Refraction and Scattering.

3.3. Important Wireless Channels Characteristics

Depending on the characteristics of the transmitted signal and the channel, fading can be either fast or slow, flat or frequency selective. This is governed by the following channel characteristics.

Channel Delay Spread: "is the time difference between the arrival of the first measured path and the last" (Liu, Sadek, Su & Kwasinski 2009: 14). If the duration of the transmitted symbol is larger than the delay spread, then copies of the first symbol arriving from multipath will overlap with the next transmitted symbol causing unpredictable amplitude and phase distortions at the receiver's side. This problem is referred to as Inter Symbol Interference (ISI) and is often mitigated by the use of Orthogonal Frequency Division Multiplexing (OFDM) which is the case in VANETs as well (see **Figure 3**).

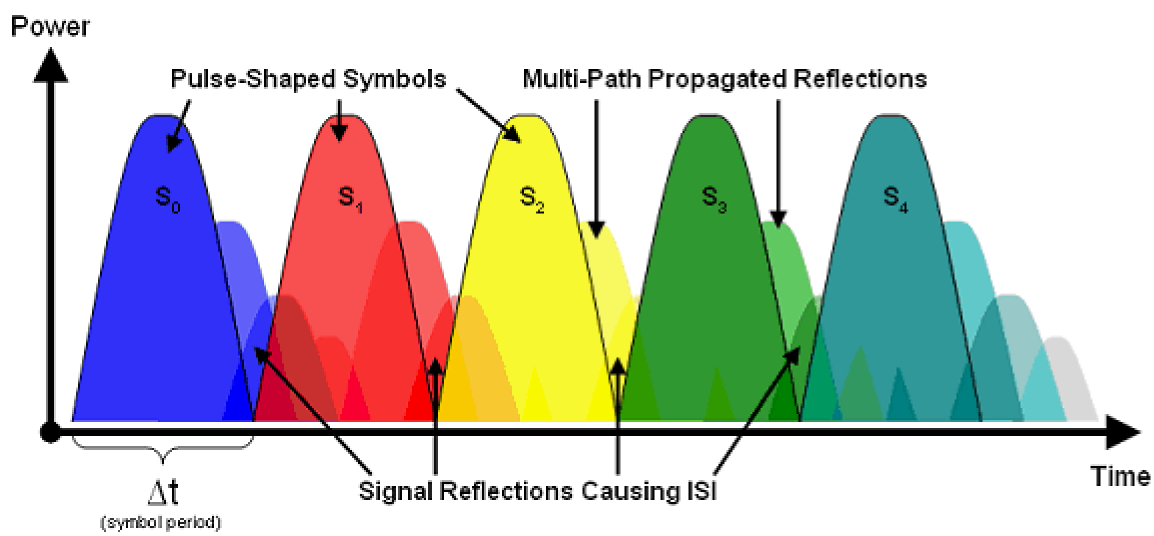


Figure 3. Inter Symbol Interference (National Instruments 2013).

Coherence Bandwidth: is the range of frequencies that can pass through the channel without phase distortion and with approximately the same original amplitude. In case the transmitted signal bandwidth is less than the channel coherence bandwidth then all the spectral components will undergo the same attenuation and linear phase shift and the channel is called a flat fading channel or a narrowband channel. Otherwise, if the coherence bandwidth is less than the transmitted signal bandwidth, different frequency ranges will suffer different attenuations and phase shifts and the channel is called a frequency selective or broadband channel. (Elmusrati 2011.)

Doppler Effect: In case of VANETs, fast relative motion speed between the sender and receiver nodes will increase the random frequency modulations appearing due to Doppler Effect of the channel. The apparent change in frequency f_d (Doppler Shift) is given in equation 11.

$$f_d = \frac{v}{\lambda} cos\, \theta \qquad (11)$$

Where θ is the angle of arrival and the velocity v is equal to the difference between the sender and receiver movement velocity. In case v is positive (i.e. the receiver is getting closer to the sender) the apparent received frequency is increased. If the receiver is getting farther from the sender, a negative shift results and the apparent received frequency is decreased. (Rappaport, Theodore 2002: 141).

“If Doppler spread is smaller than the signal’s bandwidth, the channel will be changing over a period of time longer than the input symbol duration. In this case, the channel is said to have slow fading. If the converse applies, the channel is said to have fast fading”. (Liu, Sadek, Su & Kwasinski 2009: 14.)

For a simulation scenario to be as realistic as possible, the choice of a suitable path loss model should be made carefully. Path loss models are generally classified in three categories.

Deterministic Models: Based on scientific equations and take into account several site specific variables considering its geometry and objects (buildings height, signs, trees...). They yield to more accurate results but require lots of computational resources.

Empirical Models: Approximations (curve fitting) which are based on statistical data acquired from field measurements. They are simple to implement and they have fewer parameters to adjust. However, empirical models are site specific and not always accurate.

Semi-Deterministic Models: They are based on empirical models while implementing some deterministic aspects.

3.4. Path Loss Models for VANETs

Following is the illustration of some propagation models relevant to the Vehicular Ad Hoc Networks.

3.4.1. Free Space Radio Propagation

This model was used by researchers for MANETs in the earliest studies. It assumes single unobstructed path where the signal propagates through open space without being affected by the environment. The model examines the terrain only to determine the availability of an LOS path between the sender and receiver nodes. If there is no LOS, communication is not feasible and the signal is blocked completely. The received power depends only on the transmitted power, the antenna gain and the distance between the sender and the receiver as shown in equation 12. (Singh & Lego 2011: 39.)

$$p_r = \frac{p_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (12)$$

This model treats vehicles as if they were floating points in the free space and completely ignores the effects of obstacles. Therefore it is not suitable for urban VANET scenarios.

3.4.2. Two Ray Ground Radio Propagation

This radio propagation model is widely used by MANET researchers. It takes into account the dielectric properties of the ground and considers the vehicles to be placed on a plane while assuming that the transmitter and receiver are placed on the same height. The received power is calculated as in equation 13. (Singh & Lego 2011: 39.)

$$p_r = \frac{p_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad (13)$$

Because the effect of ground reflection is proportional to the distance travelled by an electromagnetic wave, it provides considerably better results when the distance is far between the sender and receiver. (Singh & Lego 2011: 39.)

Unfortunately, Two Ray Ground Model does not consider obstacles. Furthermore, results are not accurate for short distances “because of oscillation caused by the constructive and destructive combination of the two separate paths” (Eenennaam , E.M. van 2009).

3.4.3. Ray Tracing model

Ray tracing is often used for cellular systems (such as UMTS/IMT2000) in dense urban areas because it offers high accuracy prediction. Detailed topographical information about buildings and objects is gathered including electrical properties (permittivity and conductivity) as well as its exact location and dimensions. Rays are sent out at various angles from a fixed antenna. Then for each ray, the amplitude A_n, arrival time T_n and phase θ_n are calculated using Snell's laws, the Uniform geometrical Theory of Diffraction (UTD) and Maxwell's equations. This allows the construction of a complex impulse response model which accounts for N multipath delayed components such as in equation 14. (Eenennaam , E.M. van 2009).

$$h(t) = \sum_{n=1}^{N} A_n \delta(t - T_n) \exp(-j\theta_n) \tag{14}$$

When the location of the transmitter and receiver changes, so does their surrounding environment. Thus new computations for the equation variables need to be performed. It is rather complex to apply ray tracing for a highly mobile and rapidly changing environment such as VANET. **Figure 4** represents Free Space, Two Ray Ground and Ray Tracing.

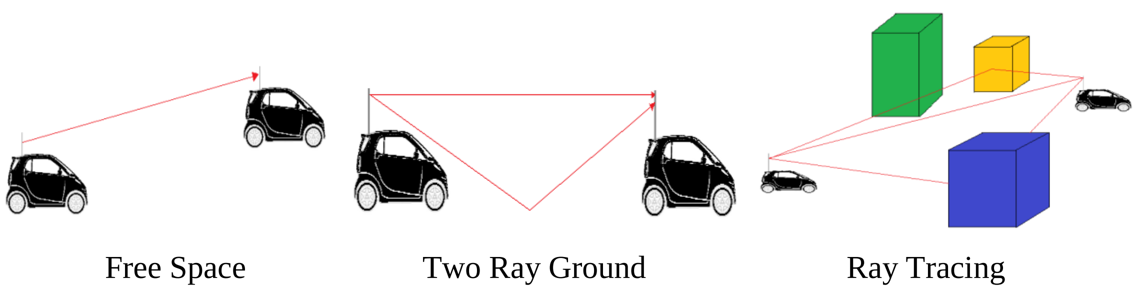


Figure 4. Free Space, Two Ray Ground and Ray Tracing path loss models

3.4.4. Log Normal Shadowing

The previous models assume ideal case of free space and calculate the received power as a deterministic function of the transmitted power and distance which is not the real case. Because of the multipath propagation effects, the received power is a random variable. The shadowing model takes that into account and consists of two parts. The first part is a path loss model to calculate the average received power and it is expressed in equation 15. (Singh & Lego 2011: 40.)

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) \tag{15}$$

Where d is the distance between the transmitter and the receiver, $p_r(d_0)$ is the mean received power at d and β is the path loss exponent which is calculated from field measurements.

The second part is Gaussian random variable which considers the fluctuations of the received power at distance d due to the multipath effect. It is denoted by X_{dB} and it has zero mean and a standard deviation σ. The overall model thus becomes:

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB} \tag{16}$$

The advantage of shadowing radio propagation model is that β and σ_{dB} can be varied to represent various different scenarios for the environment's conditions and obstacles allowing more realistic simulations to be carried out.

Table 1. Typical Values of Path Loss β (Eenennaam , E.M. van (2009).

Environment		β
Outdoor	Free space	2
	Shadowed urban area	2.7 to 5
In building	Line-of-sight	1.6 to 1.8
	Obstructed	4 to 6

Table 2. Typical Values of Shadowing Deviation σ_{dB} in dB (Eenennaam , E.M. van 2009).

Environment	σ_{dB}
Outdoor	4 to 12
Office, hard partition	7
Office, soft partition	9.6
Factory, Line-of-sight	3 to 6
Factory obstructed	6.8

3.4.5. Rician and Rayleigh fading models

These two models represent the case when several indirect paths varying in power, amplitude and phase exist between the sender and receiver. The difference between Rician and Rayleigh is that Rayleigh is a zero mean random variable while Ricean has a non-zero mean. Ricean fading model assumes the presence of a dominant component which is usually an LOS signal while Rayleigh takes into account only the multipath signals. In some of the Ricean models, the dominant wave can be expressed as the summation of two or more different signals which are out of phase. This can be useful for example, to model the Line-of-Sight plus the ground reflection. The probability density function (pdf) of a signal received over a Ricean channel can be expressed by equation 17. (Rhattoy, A. & A. Zatni 2012: 754.)

$$f(x) = \frac{2x(k+1)}{P} \exp\left(-k - \frac{(k+1)x^2}{P}\right) I_0 \tag{17}$$

Where k is the ratio of the power received in the direct LOS path, P is the average received power and I_0 is the zero-order Bessel function defined by equation 18. (Rhattoy, A. & A. Zatni 2012: 754.)

$$I_0(x) = \frac{1}{2\pi}\int_0^{2\pi} \exp(-x\cos\theta)\, d\theta \tag{18}$$

In case k=0 (no dominant path or LOS) equation 17 will be reduced to equation 19.

$$f(x) = \frac{2x}{P}\exp\left(-\frac{x^2}{P}\right) \tag{19}$$

This is the same pdf as Rayleigh distribution.

The V2V radio communication can be modeled as a Rician fading channel by setting large K-factor value as the dominant component which is often strong relatively to the multipath signals. Large buildings can be modeled as obstacles which completely block the LOS communication.

3.4.6. Nakagami Radio Propagation

Nakagami is a generic highly customizable radio propagation model. The pdf of Nakagami is given in equation 19.

$$f(x) = \frac{2m^m x^{2m-1}}{\Gamma(m)\boldsymbol{\sigma}^{2m}}\ e^{-mx^2\ /\boldsymbol{\sigma}^2}\ ,\ m \geq 0.5 \tag{20}$$

Where m is a parameter used to adjust the pdf of the Nakagami distribution to the data samples, $\Gamma(m)$ is the Gamma function and σ is the standard deviation. Rayleigh distribution can be considered as a special case of Nakagami when m = 1. To decrease fading effects greater values for m can be used. By varying its several configurable parameters, it is possible to model a wide range of scenarios ranging from perfect free space to intensely faded channels making Nakagami model well suitable for VANET simulations. (Liu, Sadek, Su & Kwasinski 2009: 22.)

4. ROUTING PROTOCOLS

Wireless transmission can be either broadcasted to a group of nodes within a specific range or forwarded towards one specific node (unicast). In case of autonomous decentralized networks, packets are forwarded using multi-hops across intermediate nodes until they reach their intended destination. The set of rules and software algorithms which determine ideal transmitting settings and the optimal path between the source and destination nodes are called routing protocols. Routing protocols performance is optimized based on specific parameters which can be for instance power consumption, packet delivery rate or end to end delay.

In VANET, the high speed of nodes movement triggers fast changes in network topologies causing frequent disconnections between nodes exchanging data. A great variation in nodes density depending on place and time of the day as well as wide range of obstacles with different shapes, sizes and surface characteristics further complicates the task of optimizing routing parameters. The design and implementation of efficient and flexible routing protocols for such a highly dynamic environment is challenging yet essential task. At first, MANET routing protocols were implement in VANETs. However, they suffered from many technical limitations. Some of those limitations are discussed in the next paragraphs.

Scalability: MANETs routing protocols consider managing a limited number of nodes. Proactive routing protocols store all the possible routes to all other nodes of the network in routing tables. In case of VANETs, a large number of nodes exist. Consequently, a huge number of valid routes are available and storing all of these becomes very costly. (Spaho, Barolli, Mino, Xhafa & Kolici 2011: 5.)

Mobility: MANET routing protocols assume completely random nodes movements. In VANET, nodes movements are constrained by the road topology and they are regulated by traffic lights and laws. An improvement of performance can be obtained by taking constrained mobility patterns into consideration. (Spaho, et al. 2011: 5.)

Intensive Use of Flooding: Most MANET routing protocols utilize flooding where every incoming packet is sent through every outgoing link except the one it arrived from. Flooding is used in reactive routing protocols for route discovery purpose until the destination is found. In order to maintain the correct route information, proactive routing protocols must keep sending control messages periodically to its neighbors or the entire network when needed. The amount of wasted bandwidth due to flooding increases greatly with a larger number of nodes causing major performance degradation. (Spaho, et al. 2011: 5.)

Localized Routing: "In proactive routing all nodes take part in building routing tables. In reactive protocols all nodes participate in the initial flooding required to find a route towards the destination" (Spaho, et al. 2011: 5).

With the large number of nodes present in VANET, it would be more efficient in terms of flexibility, control overhead and scalability to limit routing tables and route discovery to specific smaller areas (clusters). However, additional information about nodes outside that specific area needs to be provided by some location service such as GPS. (Spaho, et al. 2011: 5.)

In order for routing protocols to function effectively with large numbers of nodes in the dynamic and fast changing environment of VANETs, they need to satisfy several features.

Low Latency: To meet application requirements especially the safety applications (Spaho, et al. 2011: 5).

High Reliability: By reducing packet drop ratio and packet collisions as much as possible (Spaho, et al. 2011: 5).

Flexibility: The ability to provide the required quality of service with changing car densities, variable network area and for a wide range of different applications (Spaho, et al. 2011: 5).

Driver Behavior: The content of messages may influence driver's behavior resulting in changes in the network topology. The relation between messages and network topology needs to be considered. (Spaho, et al. 2011: 5.)

Comfort Messages: Comfort and entertainment applications are delay tolerant; routing protocols need to be designed in a way that prioritizes emergency and safety messages over comfort messages in terms of urgency and bandwidth utilization (Spaho, et al. 2011: 5).

Hierarchical Routing: Dividing the network into smaller clusters significantly reduces routing table sizes resulting in smaller overhead and lower latency in packet transmission. Unfortunately, this results in longer addresses and requires frequent updates for cluster's hierarchical addresses as the nodes are continuously moving. (Spaho, et al. 2011: 5.)

In VANET, routing protocols are classified into five categories according to their area and most appropriate applications. Those are Topology based, Position based, Cluster based, Geo cast routing protocols and Broadcast routing protocols as shown in **Figure 5**.

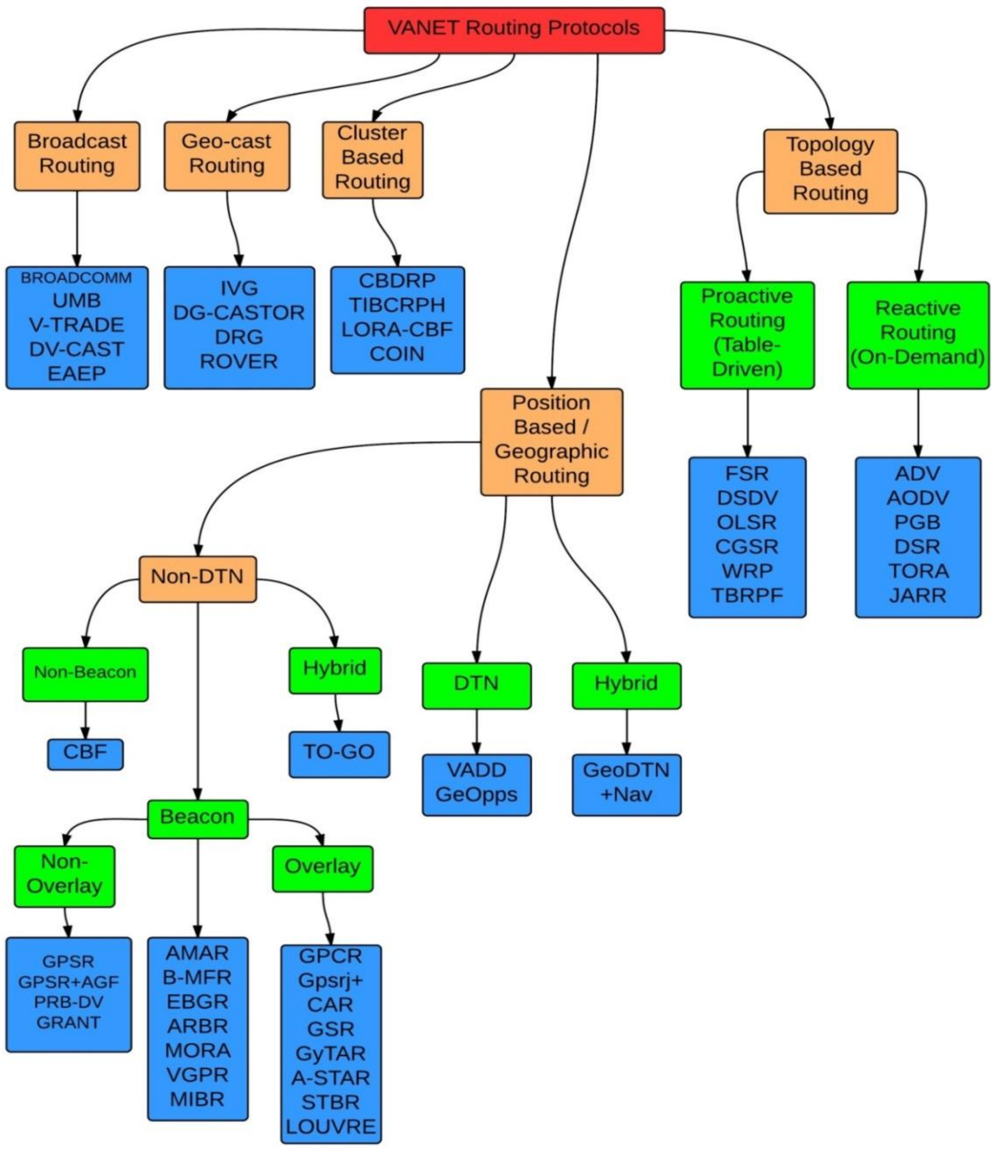


Figure 5. Classification of Routing Protocols.

4.1. Topology Based Routing Protocols

They utilize the link information within the network in order to forward data packets from source to destination. They are further classified into proactive (table-driven), reactive (on-demand) and hybrid routing protocols. (Lee & Gerla 2009.)

4.1.1. Proactive routing protocols

Information about the location of every node connected to the network is stored in tables regardless of communication requests. The table consists of entries pointing to the address of the next hop towards every possible destination whether it is needed or not. These tables are shared among neighbor nodes and updated constantly by continuously broadcasting and flooding control packets throughout the whole network. (Lee & Gerla 2009.)

Fisheye State Routing (FSR): A topology map is stored in every node and the regular link state updates are exchanged only between direct (single hop) neighbors reducing packet's overhead size and thus bandwidth consumption. Unfortunately, routing precision is reduced as the distance to the destination increases. When broadcasting link sates to farther nodes, different frequencies are used according to the destination's hope distance. Naturally, higher frequencies are used for closer nodes while the lower frequencies are preserved for the farthest ones. (Lee & Gerla 2009.)

Destination-Sequenced Distance-Vector Routing (DSDV): DSDV utilizes Bellman-Ford algorithm with the hop count as a cost variable to evaluate paths. Routing tables with entries for all nodes in the network are maintained. Updates are propagated periodically to all nodes and tagged with sequence numbers to eliminate routing loops within the network. For normal update, even sequence numbers are used. If an odd sequence number is received, it indicates an expired route thus nodes receiving this update can remove its corresponding entry from their routing tables. (Narra, Cheng & Cetinkaya 2011.)

Optimized Link State Routing Protocol (OLSR): OLSR incorporates three main mechanisms; neighbor sensing using HELLO messages, efficient control traffic flooding using Multipoint Relays (MPRs), and optimal path calculation using shortest path algorithm. Neighbors of a node can be either immediate (single hop) or two hop nodes connected through the immediate ones. They are discovered through sending periodic *hello* packets containing the sender's address and in case of two hop nodes, a list of the sender's immediate neighbors and their link status are included as well. Those packets are then stored temporarily and updated regularly. This way, each node is constantly aware of all available single and two hope neighbors around it. To avoid wasting bandwidth because of multiple duplicated transmissions, OLSR uses MPR flooding instead of normal flooding. Using two hops neighbor information, a minimum number of MPR nodes (just enough to reach every possible node in the two hop range) is saved in the MPR selector list. Then only the nodes which are registered in the source's selector list will retransmit messages thus eliminating duplicate transmissions. Every node with non-empty MPR list will periodically send topology control messages containing its address and MPR list throughout the network. Therefore, each node will have a partial topology graph of the entire network. The shortest path algorithm then uses this graph to determine optimal paths between all connected nodes. (Rastogi, Ganu, Zhang, Trappe & Graff 2007.)

Clusterhead Gateway Switch Routing (CGSR): CGSR divides the network into several clusters where each one of them is controlled by a cluster head. Then it becomes possible to apply unique coding, channel access, routing and bandwidth allocation for each cluster. (Nagaraj, Kharat & Dhamal 2011a.)

The Wireless Routing Protocol (WRP): "Each node in the network is responsible for maintaining four tables: (a) distance table, (b) routing table, (c) link-cost table, and (d) message retransmission list (MRL) table." (Nagaraj, Kharat & Dhamal 2011a).

Topology Dissemination Based on Reverse-Path Forwarding (TBRPF): Instead of transmitting full details, *hello* messages in TBRPF carry only the difference between the current and the previous network state. This reduces the size of routing messages making it possible to transmit them more frequently and thus a higher precision is achieved. (Nagaraj, Kharat & Dhamal 2011a.)

Proactive routing protocols eliminate the need for route discovery by maintaining updated and constantly available lookup tables for every possible destination. The advantage is low latency for real-time applications. Unfortunately, lots of bandwidth can be wasted maintaining some obsolete disconnected paths especially in the quickly changing topology of VANETs.

4.1.2. Reactive Routing Protocols

Reactive routing is also called on demand routing because routes between a pair of nodes are opened only when needed and maintained only as long as they are in use. The route discovery process is performed by flooding query packets into the network until a path to the destination is found. (Lee & Gerla 2009.)

Dynamic Source Routing (DSR): In DSR, Route discovery is initiated only when an unexpired route to the destination cannot be found in the source's cash. A route request packet includes a unique identification number, the address of the source node and the address of the destination node which is checked at every intermediate node traversed by the packet. If the intermediate node does not know a route to the destination it will add its address to the route record of the packet and send it to its neighbors. To reduce bandwidth consumption, route request packets are only investigated by nodes whose address is not present in the packet's route record. When the destination or an intermediate node who knows the address of destination is found, a reply is sent back to the source node using the sequence of hops already stored in the packet's route record. (Lee & Gerla 2009.)

Figure 6 illustrates the route discovery process in DSR.

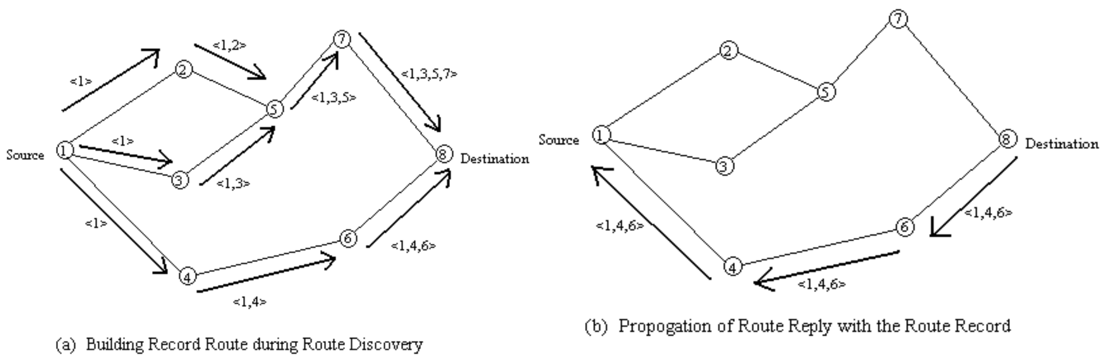


Figure 6. Route Discovery Process in DSR (Misra 2000).

One advantage of DSR over AODV comes from the fact that in DSR it is possible to allow the destination to send multiple route replies so that the source can have multiple routes to the destination. In case of low mobility networks, DSR can try the alternative routes before reinitiating the route discovery procedure gaining an edge over AODV.

Ad Hoc on Demand Distance Vector (AODV): AODV is an improved version of DSDV which burrows route discovery and maintenance techniques from DSR but differs mainly in two things. First, DSR data packets carry the whole routing information while in AODV they carry only the destination address resulting in a smaller route overhead. As the number of nodes in a network keeps increasing, the routing overhead in DSR data packets increases as well giving AODV the advantage in larger and highly mobile networks. The second difference is that in AODV, route reply packets carry the destination address and the sequence number only whereas in DSR, route reply packets carry the address of each node along the route. Because of that, DSR can find routes faster from the cache without the need to reinitiate route discovery process and performs better than AODV in smaller networks with slower changing topologies. (Rastogi, Ganu, Zhang, Trappe & Graff 2007.)

AODV+ Preferred Group Broadcasting (PGB): PGB tries to reduce redundant transmissions going through the intermediate nodes in order to reduce control messages overhead. When choosing intermediate relay nodes, very short distances result in a greater number of hops and consequently more delay and larger overhead. On the other hand, if the distance between two intermediate nodes is too far (close to the transmission range) the connection might be lost when one of them moves out of the transmission range. Another problem in ad hoc networks is the hidden terminal problem. When two A and B nodes communicate, only nodes within the transmission range of A and B will be aware and thus adjust their network allocation vector to insure not to transmit and not to interfere. Other nodes will perform the regular carrier sense and start transmitting when the medium is free. This transmission can then collide with packets being relayed through intermediate nodes between node A and B. PGB addresses the above issues by limiting the set of possible intermediate nodes. Based on the sensed signal level of the received route request, nodes are classified into three groups. Preferred group (PG) which are not too far nor too close, IN group with a signal stronger than PG (too close) and OUT group with a signal weaker than PG (too far) as shown in **Figure 7**. (Naumov, Baumann & Thomas Gross 2006.)

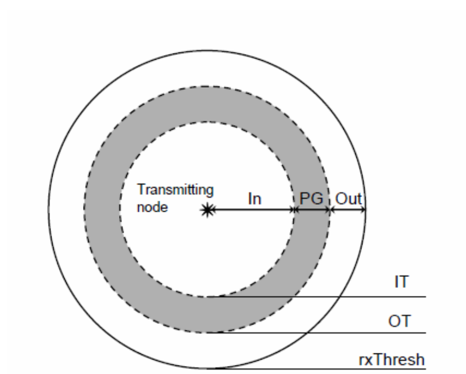


Figure 7. Node groups in Preferred Group Broadcasting (Naumov et al. 2006).

To classify a node, the power of a received signal is compared to two values; Inner Threshold (IT) and Outer Threshold (OT). Nodes are classified as IN if the received signal is stronger than IT, OUT if the received signal is weaker than OT and PG if the signal's strength falls between OT and IT. (Naumov, Baumann & Thomas Gross 2006.)

"Nodes from the PG have id = 1 and the highest priority to be chosen as relays, then nodes from the OUT group (id = 2), and finally from IN (id = 3)" (Naumov, Baumann & Thomas Gross 2006).

This solves the problem of selecting the best next hop distance. The drawbacks of using PGB are slower route discovery and interruption of broadcast in case the group is empty which might happen in sparse networks.

Temporally Ordered Routing Algorithm (TORA): TORA discovers multiple routes between the source and destination nodes. When topology changes, control messages are restricted to a small group of nodes surrounding that change thus nodes need only to maintain routing information about their adjacent neighbors. The protocol has three basic functions; Route creation, maintenance and erasure. (Hui, & Datta 2012.)

Route creation starts by setting the propagation ordering parameter (also called the height) to 0 for the destination and Null (unidentified) for all other nodes. The source then sends a query packet containing the destination's id. Nodes with identified height will reply sending their height in an update packet. Upon receiving an update packet, the receiving node will set its height to be equal to one plus the packet's sender. A node is said to be upstream if it has a greater height and downstream if it has a lower height. This produces a Directed Acyclic Graph (DAG) connecting the source to the destination. (Hui, & Datta 2012.)

Route maintenance is carried out when a node loses every possible route to a destination due to the detection of a link failure or link reversal after receiving an update packet. To fix the problem, every node will send a packet to reverse the links of its neighbors who do not have any ongoing links. The neighbors then send the same kind of packet to their neighbors and so on until each node has at least one outgoing link. One disadvantage of TORA is that route maintenance can sometimes produce excessive overhead causing network congestion. (Hui, & Datta 2012.)

In case a node wants to communicate but exists in a network partition where all nodes in that partition do not know a route to its intended destination, it initiates a route erasure process by flooding clear packets throughout the network. Upon receiving a clear packet, nodes will set the links to their neighbors to unassigned. Clear packets keep flooding throughout the network until all the invalid routes to the inaccessible destination are erased. (Hui, & Datta 2012.)

TORA has the advantage of providing each node with a route to every other node connected to the network while minimizing control messages. Unfortunately, its route maintenance is resource consuming especially in highly dynamic VANET networks. (Hui, & Datta 2012.)

By performing route discovery only when needed, reactive routing protocols can save lots of bandwidth compared to proactive routing protocols. Unfortunately, this comes at a cost of high end to end delay. Furthermore, the use of excessive flooding for route maintenance may cause interruption of node communication. (Hui, & Datta 2012.)

4.1.3. Hybrid Topology Based Routing Protocols

Hybrid protocols try to combine techniques from both active and reactive routing protocols to achieve reduced initial route discovery delay time while maintaining small packet overhead.

Zone Routing Protocol (ZRP): The network is split into several zones where each of these zones contains a group of nodes confined inside the radius length. If the destination is another node inside the same zone, a proactive intra-zone routing protocol (IARP) is used. Otherwise, a reactive inter-zone routing protocol (IERP) will first forward the data to the intended zone before it is delivered to its final destination. (Nagaraj, Kharat & Dhamal 2011a.)

Hybrid Ad Hoc Routing Protocol (HARP): The main difference between ZRP and HARP is that in HARP the network is divided into non-overlapping zones where communication stability is a priority. "It is not applicable in high mobility ad-hoc networks" (Nagaraj, Kharat & Dhamal 2011a).

4.2. Position Based Routing

In Geographical (position based) routing protocols, Nodes are assumed to have access to position determining services such as GPS which provide them with information about their own position as well as the destination's position. As for the one hop neighbors, they are discovered through beaconing in a periodic manner to avoid collision. Therefore, there is no need to use routing tables neither to exchange link state information. Routing decisions are made mainly based on the packet's destination and the position of the one hop neighbors. Position based routing protocols are classified into three categories based on the way they handle network disconnections. These are non-Delay Tolerant Networks (non-DTN), Delay Tolerant Networks (DTN), and hybrid. (Lee & Gerla 2009.)

4.2.1. Hybrid Position Based Routing

GeoDTN+Nav: A hybrid routing protocol which implements three different routing modes; the greedy mode, perimeter mode and DTN mode. GeoDTN+Nav detects the quality of network connections by measuring the number of hopes traversed by a packet, neighbor's delivery ratio and direction related to the destination. This data is provided by hardware equipment such as an Event Data Recorder (EDR) and navigation systems and is then fed to a virtual navigation interface to select its routing mode and forwarder. (Lee & Gerla 2009.)

Figure 8 illustrates a virtual navigation interference.

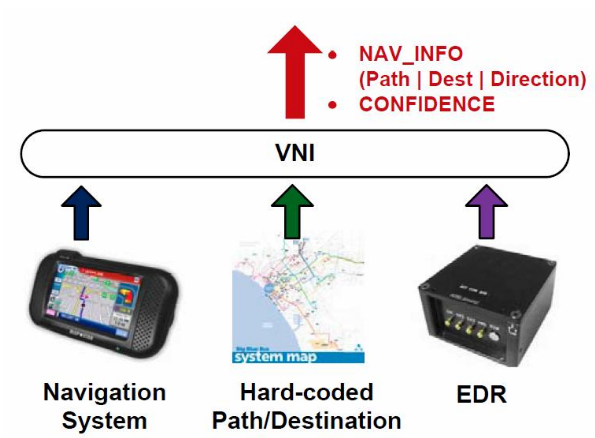


Figure 8. Virtual Navigation Interface (Lee & Gerla 2009).

4.2.2. Delay Tolerant Network (DTN) Position Based Routing

VANETs are highly mobile wireless networks suffering from frequent disconnections. Using delay tolerance and allowing nodes to store packets and carry them until a new path becomes available to the destination can be advantageous. (Lee & Gerla 2009.)

Vehicle-Assisted Data Delivery Routing Protocol (VADD): At intersections, VADD evaluates every branched road based on cars density, average car speed and segment length and tries to pick the path with minimum delay. After selecting a path, different variations of VADD select the next forwarding node differently. Location First Probe (L-VADD) selects the closest node to the forwarding path regardless of its movement direction. On the contrary, direction First Probe (D-VADD) choses a node moving in the direction of the selected path regardless of how far it is positioned. Multi-Path Direction First Probe (MD-VADD) selects several nodes heading towards the selected path in order to be able to take advantage of the one offering shortest delay. Hybrid Probe (H-VADD) combines L-VADD and D-VADD to reduce D-VADD's large delay and mitigate L-VADD's routing loops. "Results comparing with GPSR plus buffer and various versions of VADD show that H-VADD has the best performance". (Lee & Gerla 2009.)

Geographical Opportunistic Routing (GeOpps): GeOpps obtains suggested paths for cars from their equipped navigation system. It predicts the arrival time of a packet by calculating the shortest distance between the destination and the nearest point on the path. (Lee & Gerla 2009.)

Since GeOpps exploits other nodes navigation information, privacy might be an issue.

In **Figure 9**, Node A selects the node travelling on route N1 since it offers a closer nearest point to the destination compared to N2.

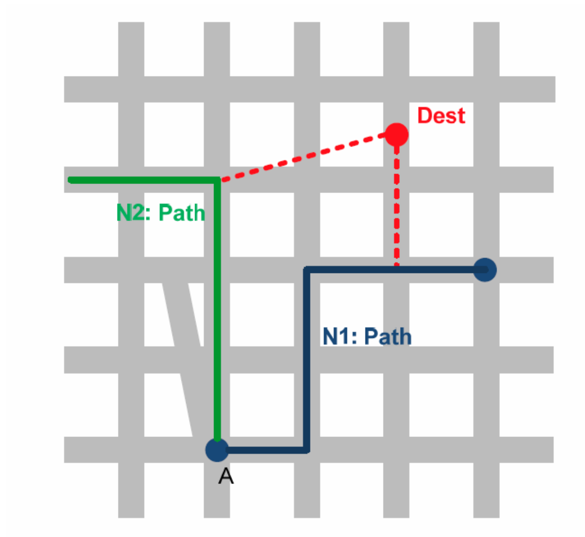


Figure 9. Nearest Point Calculations in GeoOpps (Lee & Gerla 2009).

4.2.3. Non-Delay Tolerant Position Based Routing

Non-DTN position based routing can use beacons, be non-beacon or a hybrid of the two.

4.2.3.1. Non-Delay Tolerant non-Beacon

Contention-Based Forwarding (CBF): In CBF, the source transmits packets to all of its neighbors and leaves the decision of the next hop to them. By comparing its own distance to the destination versus the source's distance to the destination, the node with the shorter distance makes the decision to forward the packet and suppresses other nodes preventing them from broadcasting. As a result CBF does not require the use of proactive beacon messages. (Lee & Gerla 2009.)

4.2.3.2. Non-Delay Tolerant Hybrid

Topology-assist Geo-Opportunistic Routing (TO-GO): The destination node usually exists in a different street than the source node and packets need to travel several intersections before arriving. A target node is selected by a greedy algorithm based on information gathered from the two-hop beaconing procedure. Unlike CBF, packets are forwarded towards the selected target node instead of the destination. (Lee & Gerla 2009.)

4.2.3.3. Non-Delay Tolerant Beacon

Receive on Most Stable Group-Path (ROMSGP): In most cases, disconnection happens when a vehicle moves out of the transmission range of its neighbor. ROMSGP classifies cars into four groups according to their velocity vector and considers routing to be stable when the pair of nodes belongs to the same group. In **Figure 10** there are two possible routes to forward a packet from node A to node B; route A–B–D–E and route A–C–D–E. However, node B is more likely to move out of the transmission range of node A making route A–C–D–E a more stable choice. (Nagaraj, Kharat & Dhamal 2011a.)

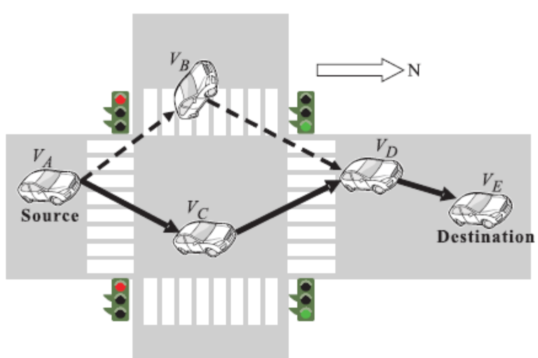


Figure 10. Choosing a route in ROMSGP (Lin, Chen And Lee 2010: 918).

Adaptive Movement Aware Routing (AMAR): In AMAR, every vehicle calculates its own position and velocity vector based on data obtained from GPS. Priorities are then assigned to neighbors in order of a weighted score W_i calculated from their speed, direction and position as follows:

$$W_i = \alpha P_m + \beta D_m + \gamma S_m \tag{21}$$

Where α, β and γ are the weight of the three used metrics P_m, D_m and S_m representing respectively the position, direction and speed of a neighbor node, $\alpha + \beta + \gamma = 1$.

"This scheme is suitable for highly mobile vehicular ad hoc network and even it performs better in case of pure greedy forwarding failure" (Raw & Das 2011: 440).

Border-Node Based Most Forward within Radius Routing Protocol (B-MFR): "In this method, a packet is sent to the border-node with the greatest progress as the distance between source and destination projected onto the line drawn from source to destination" (Raw & Das 2011: 440). This helps avoid unnecessary retransmission of packets through nodes within the transmission range. M-MFR forwarding method is shown in **Figure 11**.

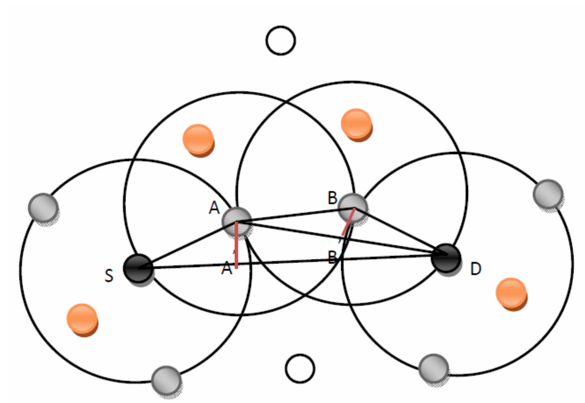


Figure 11. B-MFR Forwarding Method (Raw & Das 2011: 440).

Edge Node Based Greedy Routing Protocol (EBGR): During packet transmission three methods are used. Neighbor node selection method collects data from all neighbors within transmission range. Node direction identification method identifies other node's velocity vector relatively to the destination. Finally, edge node selection method selects farthest node within transmission's range as the next hop. EBGR's advantages are minimum number of hops and maximum possible network throughput. (Raw & Das 2011: 440.)

The Associativity-Based Routing (ABR): ABR uses the concept of associativity between nodes when selecting the optimal path for transmission. Associativity is evaluated based on the number of beacons received by a mobile host form its neighbor nodes. A low number of received beacons indicates a highly mobile node while a high number of received beacons indicates a node in low mobility or stable state. Mobile hosts in stable state are ideal to be chosen as ad-hoc relays. (Taleb, Sakhaee, Jamalipour, Hashimoto, Kato & Nemoto 2007.)

Vertex-Based predictive Greedy Routing (VGPR): It discovers a multi-hop sequence of usable junctions from the source to a fixed infrastructure then forwards packets using a greedy scheme. The evaluation and selection of junctions is done based on vehicles position, velocity and trajectory. (Nagaraj, Kharat & Dhamal 2011a.)

Dynamic Time-Stable Geocast Routing (DTSG): Designed to provide improved performance in low density networks by dynamically adjusting itself based on the network's vehicles speed and density. DTSG has two phases, pre-stable for local message dissemination and stable for storing and forwarding messages. (Nagaraj, Kharat & Dhamal 2011a.)

Greedy Perimeter Stateless Routing (GPSR): GPSR has two packet forwarding modes; greedy forwarding mode and perimeter forwarding mode. By default, greedy mode is used where packets are always forwarded to the neighbor who is closest to the destination until the destination is reached. This strategy can fail if no neighbor is closer to the destination than the node itself and the packet is said to have reached a local maximum. To recover from this problem, the perimeter forwarding mode is applied using the right hand rule.

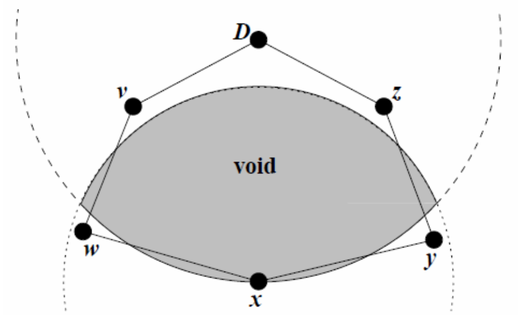


Figure 12. Node x's void with respect to destination D (Karp & Kung 2000).

In **Figure 12**, two nodes w and y are inside the transmission range of *x*. Unfortunately, they are both farther to D than *x* therefore transmission fails in greedy mode and the node *x* enters recovery mode using perimeter forwarding in order to route around the shaded area (called the void). The next hop is then selected using right hand rule. At first, the next hop is the first node *w* positioned anticlockwise to the edge connecting the source *x* to the destination *D.* Afterwards, the first node positioned anticlockwise to the edge connecting the current node which is holding the packet *w* and its source node *x* will be selected and that would be *v.* Applying right hand rule will result in the sequence *x* → *w* → *v* → *D* successfully delivering the packet to its intended destination. (Karp & Kung 2000.)

GPSR + Advanced Greedy Forwarding (AGF): When applying GPSR in highly dynamic networks such as VANETs, is suffers from two shortcomings. First, the next hop might have outdated information of its own neighbors' position. The second problem is that the destination's location in the packet's header is never updated while the packet is being forwarded from the source to the (mobile) destination. AGF includes the speed and direction of the sender node in beacon packets as well as the packets total travelling time. Performing mathematical calculations, a node can filter out neighbors who are expected to leave its transmission range by the time it receives a packet. Knowing total travelling time, each forwarding node is able to estimate the updated position of the destination. (Lee & Gerla 2009.)

Position-Based Routing with Distance Vector Recovery (PBR-DV): When transmission fails in greedy mode, the node at local maximum will broadcast its own position and the destination's location in a request packet. If the node receiving the request packet is closer to the destination it will reply. Otherwise, it will append the source's address to the packet and broadcast it again. This insures that upon receiving the reply packet, the node suffering from maximum local will be provided with a full sequenced route to a node closer than itself to the destination and thus it can resume broadcasting in greedy mode. (Lee & Gerla 2009.)

Greedy Routing with Abstract Neighbor (GRANT): In GRANT, nodes know the location of its multi-hop neighbors. This helps the node to make better routing decisions and increases its chances to avoid falling into a local maximum. (Nagaraj, Kharat & Dhamal 2011a.)

Those routing protocols operate on a set of nodes overlaid at strategic positions on top of the network such as street intersections where packets make turns towards different road segments.

Greedy Perimeter Coordinator Routing (GPCR): In urban areas, city streets form a natural planner graph. GPCR takes advantage of that characteristic by trying to forward messages to nodes at intersections without the need of using street maps. (Nagaraj, Kharat & Dhamal 2011a.)

“GPCR traverses the junctions by a restricted greedy forwarding procedure, and adjusts the routing path by the repair strategy which is based on the topology of streets and junctions” **(**Lin, Wei , Chen & Lee 2010).

GpsrJ+: In GpsrJ+ nodes use two-hop beacons to figure out to which road segment their neighbors are going to forward a relayed packet. Packets are forwarded to neighbors who would send it through a road segment in a different direction. If such nodes are not found, junctions are neglected and packets are simply forwarded to the farthest neighbor node thus eliminating the unnecessary stop at junctions while maintaining the efficient planarity of topological maps. “GpsrJ+ manages to increase packet delivery ratio of GPCR and reduces the number of hops in the recovery mode by 200% compared to GPSR.” (Lee & Gerla 2009.)

Figure 13 compares routing techniques of Gpsr+ and GPCR.

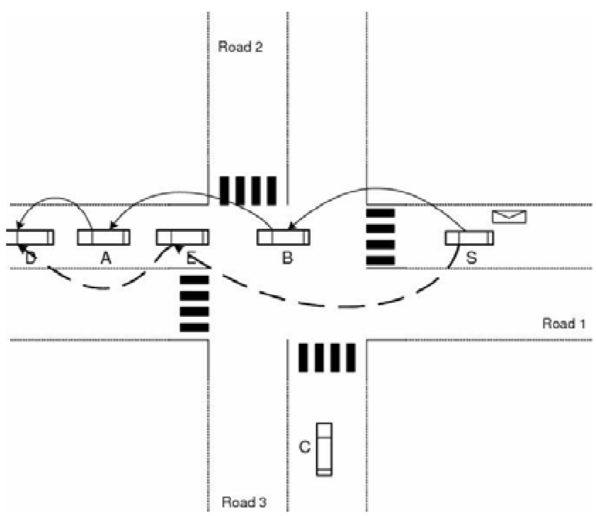


Figure 13. Gpsr+ versus GPCR (Lee & Gerla 2009).

Connectivity Aware Routing Protocol (CAR): CAR uses AODV+PGB techniques for route discovery but records different data into the packet's header. Upon receiving a path discovery packet, a node existing at a junction will consider itself an anchor point if its traveling trajectory is not parallel to the trajectory of the sending node. In case the destination receives several path discovery packets, it will pick the one with the best connectivity and minimum delay. AGF then forward the route reply through the anchor points which are recorded in the chosen path discovery packet. (Lee & Gerla 2009.)

Diagonal-Intersection-based Routing Protocol (DIR): DIR is an improvement over CAR. It forwards packets throughout a sequence of diagonal junction nodes connecting the source to the destination. Between every pair of diagonal junction nodes, there exist several sub-paths connecting them. DIR dynamically adjusts itself to the network condition by computing and selecting the sub-path which offers the minimum delay. DIR often utilizes less anchor points between source and destination resulting in a better performance compared to CAR. **(**Lin, Wei , Chen & Lee 2010.)

Figure 14 compares CAR to DIR in terms of packet routing.

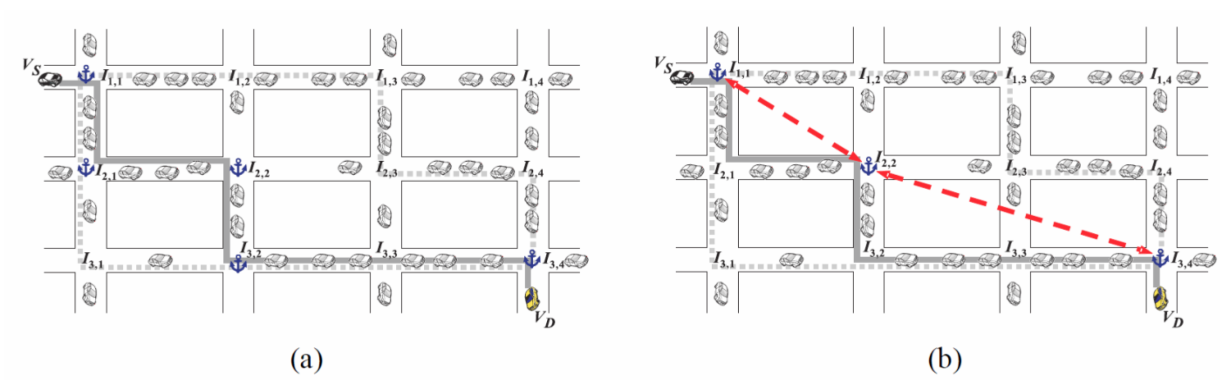


Figure 14. Comparison of Packet Routing Between DIR and CAR (Lin, Chen & Lee 2010: 917).

Geographic Source Routing (GSR): Unlike CAR, a map is used in GSR to compute the Dijkstra’s shortest path while considering junction nodes as the vertices and streets connecting those vertices as the edges. Packets are forwarded in greedy mode throughout vertices. (Nagaraj, Kharat & Dhamal 2011a.)

Anchor-Based Street and Traffic Aware Routing (A-STAR): Similar to GSR, A-STAR forwards packets through anchor points using Dijkstra algorithm. However, as the name implies, traffic is taken into consideration when deciding which anchor points form the shortest path. When computing the shortest path, the protocol relies on two kind of maps generated from data provided by the roadside deployment units. Statically rated maps which show bus routes usually provide a stable amount of traffic resulting in connected paths. Dynamically rated maps are generated according to the road’s real time conditions thus providing better accuracy. When a packet cannot be forwarded because of a disconnected path, the node will re-compute a new anchor path and notify the network about the out of service path to prevent other packets from falling into the same situation. (Lee & Gerla 2009.)

Street Topology Based Routing (STBR): At every junction one master node is selected to be responsible for checking link states to other junctions. Link information about direct and two-level junction nodes are stored and exchanged between those master nodes. STBR then forwards packets depending on their geographical distance to the street where the destination exists. (Lee & Gerla 2009.)

Greedy Traffic Aware Routing Protocol (GyTAR): Relying on data provided by roadside units, a node evaluates its neighbor junctions based on two configurable parameters; their traffic density and distance from destination. The resulting score is then used to decide the next hop at every junction. GyTAR implements greedy algorithms for packets forwarding. (Lee & Gerla 2009.)

Geographical (position based) routing utilizes information provided by navigation systems to eliminate the need for exchanging link state information and keeping established routes. It provides a robust and easily scalable connectivity with less disconnections and wasted bandwidth when compared to topology based routing. The only disadvantage for position based routing protocols is that they require continuous availability of position determining services (such as GPS) in every node which might not work in some places (such as tunnels) where satellite signals cannot reach the cars.

4.3 Cluster Based Routing Protocols

The network is divided into several clusters where each one them consists of member nodes and a cluster head. The cluster head is then responsible for delivering incoming packets to the members and to forward outgoing packets to other cluster heads in order to be delivered to one of their members. This approach provides better scalability for large networks but causes longer delays and larger packet overhead.

Cluster-Based Directional Routing Protocol (CBDRP): CBDRP is planned for highway scenarios. Vehicles with the same moving direction are categorized into one cluster and head selection and maintenance takes into account the velocity and direction of vehicles movement. Packets are sent from the source node to its cluster head. The cluster head then forwards the packet to the head of the destination cluster which in turn delivers the packet to the target node. Head selection and maintenance and routing procedures are shown in **Figure 15** and **Figure16** respectively. (Xia, Song & Shen 2010.)

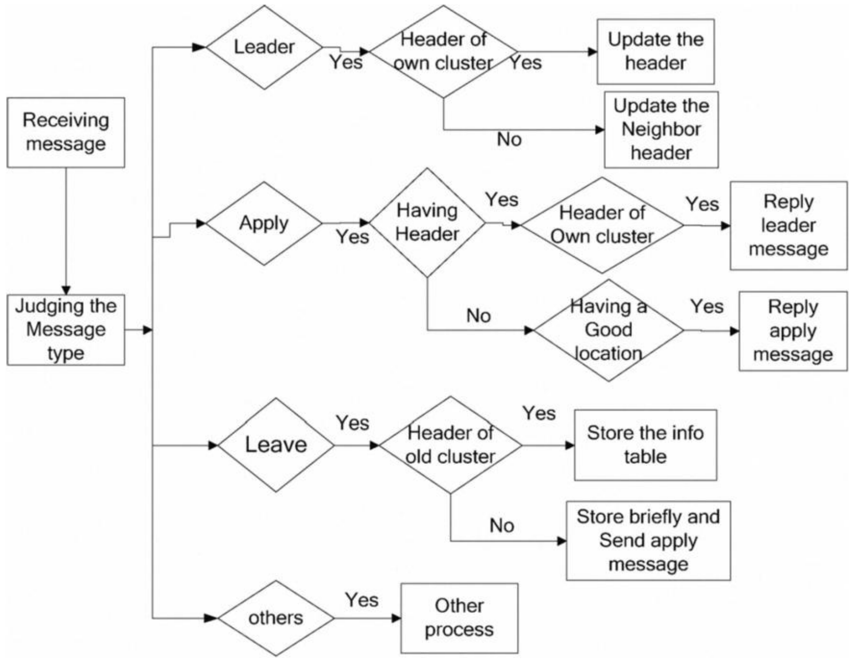


Figure 15. Head Selection and Maintenance in CBDRP (Xia, Song & Shen 2010).

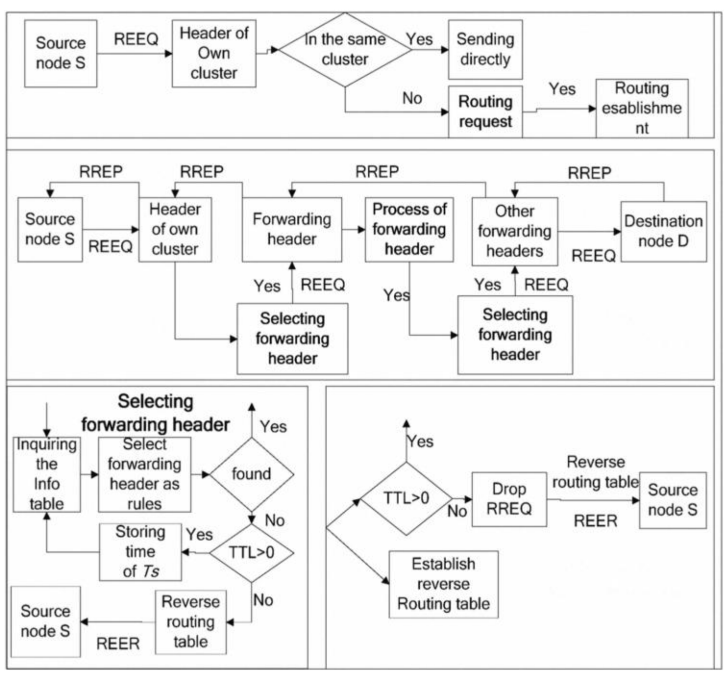


Figure 16. The Routing Procedure in CBDRP (Xia, Song & Shen 2010).

According to the simulation results of (Xia, Weiwei, Tiecheng Song & Lianfeng Shen 2010), “The CBDRP is superior to AODV and GPSR protocols, can provide high link stability, high packet delivery ratio and small latency for safety application to match the highway applications”.

Location Routing Algorithm with Cluster-Based Flooding (LORA-CBF): Each cluster has one head and one or more gateways responsible for communications with other cluster heads. Cluster tables containing the addresses of members and gateway nodes are stored within the cluster head. If the source knows the address of the destination it will start forwarding the packet immediately. Otherwise, the packet is kept in the source’s buffer and location request packets are broadcasted. These packets are retransmitted solely by cluster heads and gateways belonging to other clusters until they reach the head of the cluster where the destination is located. The cluster head then sends a location reply packet back to the source using geographical routing. (Santos, Álvarez & Edwards 2005.)

Clustering for Open IVC Network (COIN): COIN picks the node which has the lowest relative mobility to other cluster members and makes it the cluster head. This insures more robust radio connections lasting for as long as possible. (Nagaraj, Kharat & Dhamal 2011a.)

4.4. Geo Cast Routing Protocols

Geo cast routing protocols use location based multicast routing and tries to distribute packets from the source node to all other nodes which exist within a defined geographical area called ZOR (Zone of Relevance). (Nagaraj, Kharat & Dhamal 2011a.)

Inter-Vehicle Geocast (IVG): IVG is implemented in highway collision avoidance. A damaged car would broadcast an alert to a multicast group of vehicles inside a risk zone. This risk zone is defined according to the damaged car location and the driving paths which may be affected by it. Cars receiving the alert message will determine if they are in danger depending on their moving trajectory relative to the car in accident location then rebroadcast the alert message using the farthest available car as a relay. (Allal & Boudjit 2012.)

Figure 17 illustrates the relay selection procedure in IVG.

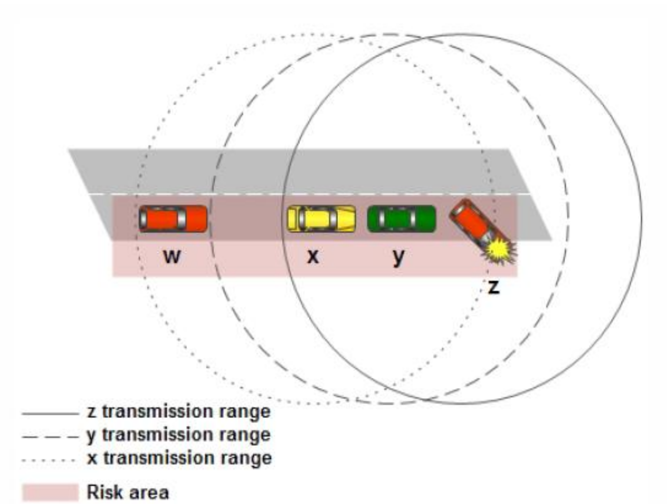


Figure 17. IVG Relay selection (Allal & Boudjit 2012: 324).

Direction-based GeoCast Routing Protocol for query dissemination in VANET (DG-CASTOR): This routing protocol tries to predict which neighbors will be able to maintain their link connection active with the sender node during a given period of time. A relevant Rendezvous zone is defined according to the relative speed and trajectory of the sender node and its neighbors. (Allal & Boudjit 2012.)

Distributed Robust Geocast (DRG): The zone of relevance (ZOR) is defined based on geographical criteria and the zone of forwarding (ZOF) is defined as the set of nodes allowed to forward messages. When selecting relays, farther nodes are preferred. When a node receives a message, it tests its relevance according to its geographical location. Nodes belonging to the ZOF will forward the message while nodes belonging to the ZOR will read it. If the node is outside of both the zone of relevance and forwarding then it will drop the message. (Allal & Boudjit 2012.)

Robust Vehicular Routing (ROVER): ROVER is used to deliver messages produced by applications to all nodes belonging to the ZOR. It limits broadcasting to only control packets while data packets are forwarded using unicast. Rover assumes that every car has its own identification number and onboard GPS receiver allowing access to digital maps. ZOR is assumed to have a rectangular shape and ZOF is defined to include all nodes from ZOR as well as the sender node. (Allal & Boudjit 2012.)

Dynamic Time-Stable Geocast Routing (DTSG): The main function of DTSG is to alert cars on highways about important events such as accidents. DTSG assumes that vehicles travel in form of groups with equal speed along the highway. Cars travelling in the opposite direction are called Helping Vehicles. Messages can be transferred through those helping vehicles to reach other groups of cars. (Allal & Boudjit 2012.)

4.5. Broadcast Based Routing Protocols

Broadcast routing is implemented in various VANET applications to broadcast information such as weather, road conditions, advertisements and announcements.

BROADCOMM: The highway is split into cells and the cars are categorized into two levels of hierarchy. The first level contains all the nodes within the cell. Some of the nodes which are positioned near the geographical center of cells are selected as cell reflectors. For certain intervals of time, cell reflectors perform the duty of a cluster head forwarding emergency messages from cell members towards neighbor cells. “This protocol performs similar to flooding base routing protocols for message broadcasting and routing overhead” (Nagaraj Uma & Poonam Dhamal 2011b.)

Urban Multihop Broadcast Protocol (UMB): UMB provides solutions for the broadcasting storm, hidden node and reliability problems found in multi-hop broadcasting techniques. The street part within the transmission range of a source is divided into segments and only a single node which is located in the farthest non empty segment is allowed to forward packets thus eliminating the need for network topology information and utilizing the network more efficiently. Upon receiving a message the node will reply with acknowledgement packet before forwarding to insure reliability. To avoid the hidden node effect, a Request to Broadcast and Clear to Broadcast handshakes are implemented between the two communicating nodes. Repeaters are installed at intersections to disseminate arriving messages into all directions. (Lopez , Jesus Gabriel Balderas 2010.)

Vector Based Tracing Detection (V-TRADE): V-Trade is similar to ZRP; it relies on GPS service to categorize neighbor nodes into different forwarding groups based on their location and movement direction. It allows only specific nodes from every group to resend the messages. “V-TRADE improves the bandwidth utilization but some routing overheads are associated with selecting the next forwarding node in every hop”. (Nagaraj Uma & Poonam Dhamal 2011b.)

Distributed vehicular broadcast protocol (DV-CAST): Neighbors are classified into three groups based on the connection quality. For well-connected neighbors, a persistence scheme is used which can be weighted p, slotted 1 or p persistent. For sparsely connected neighbors, it is possible to rebroadcast to nodes travelling in the same direction. In case of totally disconnected neighbors, the car will store the message waiting for a new node to enter its transmission range unless the expiry time is reached. Then it will discard the message. (Nagaraj Uma & Poonam Dhamal 2011b.)

5. SURVEY OF VANET SIMULATORS

Deployment and field testing of Vehicular ad-hoc networks is highly costly and requires intensive labor. A practical alternative is the use of simulation software to evaluate the performance in a wide variety of scenarios prior to the actual implementation.

The difficulty of VANET simulation comes from the need for simulating mobility patterns as well as network and data transmission simultaneously. This can be accomplished by using first a mobility generator such as SUMO to generate mobility traces and feed those traces to a network simulator such as ns-2. There exist also several tools with graphical user interfaces such as MOVE which can help the researcher to easily generate trace files from mobility generators without the need of programming skills. An integrated VANET simulator such as NCTUns offers both a mobility generator and a network simulator in one package. Integrated simulators offer better flexibility when adjusting parameters and modifying the simulation scenarios because the results of these modifications can be observed immediately. **Figure 18** illustrates a classification of VANET simulators.

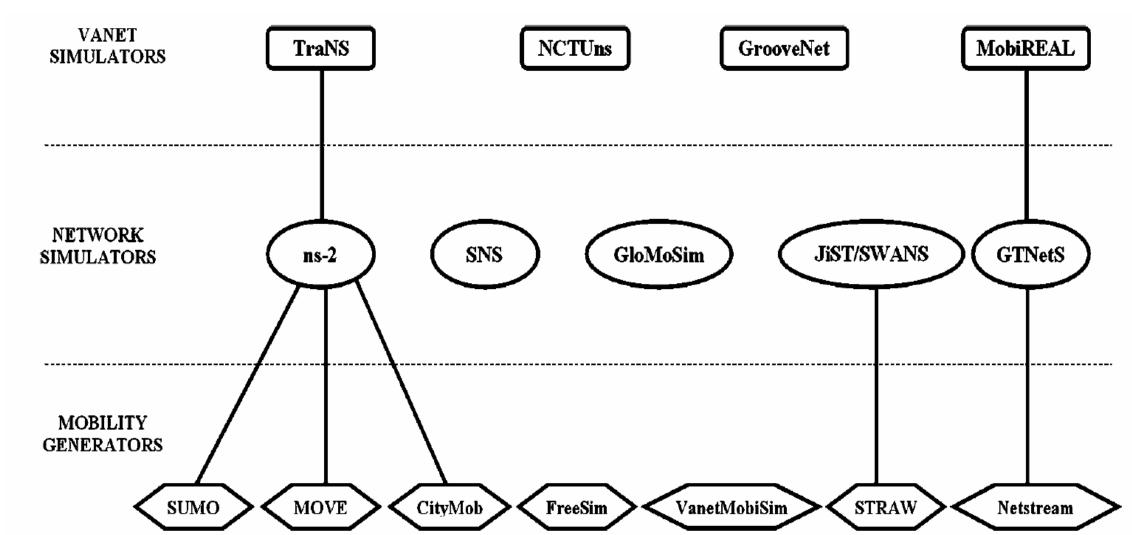


Figure 18. Taxonomy of VANET Simulators (Martinez, Toh, Cano, Calafate & Manzoni 2009)

5.1. Mobility Generators

5.1.1. Simulation of Urban Mobility (SUMO)

SUMO is an open source, microscopic, multi-modal traffic simulator. It is purely microscopic; each car is modeled separately to follow its own route moving individuality throughout the network. It supports both continuous and time discrete car movement for different types of cars. It is capable of simulating multilane-streets with lane changing, traffic lights and multiple traffic rules. According to their official website; network size is limited to 10000 streets and simulations are as fast as 100000 vehicle updates per second when running on a 1GHz processor. A set of additional tools are downloadable through their website to help import, process and export data. As a result, SUMO is able to import several different network topology descriptions and map file formats including the widely used Shapefiles, OpenStreetMaps (OSM) and XML-Descriptions. (SUMO 2013.)

5.1.2. Mobility Model Generator for Vehicular Networks (MOVE)

MOVE is built on top of SUMO. It offers graphical user interfaces to help generate realistic mobility traces rapidly thus saving users the trouble of writing simulation scripts. The road map editor allows the user to manually design roads or import them from either Google Earth or the Topologically Integrated Geographic Encoding and Referencing system (TIGER) maps. It is also possible to automatically generate a completely random map. Using the Vehicle Movement Editor, automatic car movement can be defined by setting parameters for each vehicle such as acceleration, max speed, and turning probability at intersections. Data collected by MOVE is sent to the SUMO compiler to generate a trace file. Then, using the Traffic Model Generator for VANET tool, it is possible to easily configure and run a simulation scenario in either ns-2 or Qualnet. (LENS 2012.)

Figure 19 shows the menus of MOVE's mobility model generator and traffic model generator

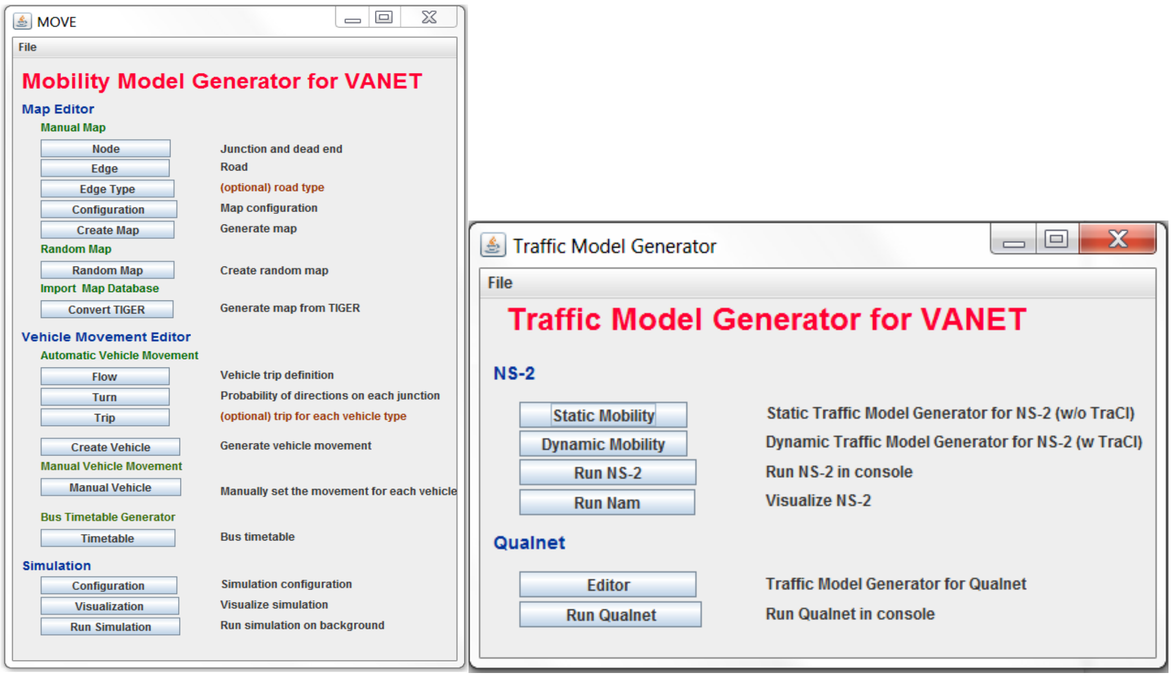


Figure 19. a) MOVE Main GUI. b) MOVE's Traffic Generator.

5.1.3. VanetMobiSim

Is an open source mobility generator based on CanuMobiSim's architecture. It supports both micro and macro mobility models and can take into account the interaction between them when producing vehicular movement traces. (Fiore, Harri, Filali & Bonnet 2007.)

Macro Mobility Features: The road topology can be defined manually from the user by listing the vertices of the graph and their interconnecting edges. It can be also imported from Geographic Data Files (GDF), the TIGER files for USA maps or generated randomly. A useful feature in VanetMobiSim is the possibility of defining areas with different and customizable car densities which are also known as clusters. The simulator supports multiple lanes, can assign different speed limits for each road in the topology and supports stop sign and traffic lights at intersections. The trip generation module supports two options; random trip generation in which only the start and end points are defined. The second option is activity sequences generation. It relies on Dijkstra's algorithm to select shortest path with edges cost inversely proportional to their length. It is also possible to take the traffic congestion level as well as speed limits of the different roads into consideration when planning the road trip with activity sequences generation. (Fiore, Harri, Filali & Bonnet 2007.)

Micro-Mobility Features: The variation of individual car speeds and acceleration can be computed deterministically using the Graph-Based Mobility Model (GBMM), the Constant Speed Motion (CSM) or the Smooth Motion Model (SMM). The Fluid Traffic Model (FTM) and Intelligent Driver Model (IDM) are also supported to account for the effect of nearby cars movement on the driver's behavior. VanetMobiSim introduces two new original microscopic mobility models which extend the IDM model. The Intelligent Driver Model with Intersection Management (IDM-IM) adds intersection handling capabilities for the driver's behavior. The second model is named Intelligent Driver Model with Lane Changes (IDM-LC) and it adds the possibility of changing lanes and overtaking cars to the IDM model. (Fiore, Harri, Filali & Bonnet 2007.)

5.1.4. Street Random Waypoint (*STRAW)*

According to the author's website, STRAW implements mobility models based on real vehicular traffic data obtained from US cities. It 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. The current implementation is written exclusively for the discrete-event simulator JiST/SWANS and cannot produce trace files which are directly usable by other network simulators. A tool for converting TIGER maps into a format usable by STRAW is available for download at their website. (Bustamante 2013).

5.1.5. FreeSim

FreeSim is developed by Dr. Jeffrey Miller at the University of Alaska, Anchorage as a macroscopic and microscopic traffic simulator. Freeway systems are represented as a graph data structure with edge weights determined by the current speeds. Traffic algorithms can be implemented for either the whole network or specific individual nodes. Input data can be manually generated by the user or converted from actual data provided by transportation organizations. (Martinez, Cano, Calafate & Manzoni.)

5.1.6. CityMob

CityMob is designed specifically to allow researchers to easily model car accidents and to test flooding based alert protocols used to help other cars to avoid the damaged cars and traffic jams because of accidents. It is developed targeting best possible compatibility with the ns-2. It supports three mobility models; simple model, Manhattan model and realistic downtown model. In the simple model, vehicles move in straight lines vertically or horizontally without changing their direction. In the Manhattan model, the city is divided into uniformly sized blocks. Streets are all two-way with single lane and cars move in random directions. Semaphores are implemented at random positions and with random delays to simulate car stoppage. In the Downtown model, zones with higher cars density can be defined by their (x, y) coordinates as long as they do not exceed 90% of the total area of the map. Those areas represent the downtown and cars move slower inside them compared to the rest of the map. (Martinez, Cano, Calafate & Manzoni.)

Figure 20 shows Citymob's GUI along with an example scenario.

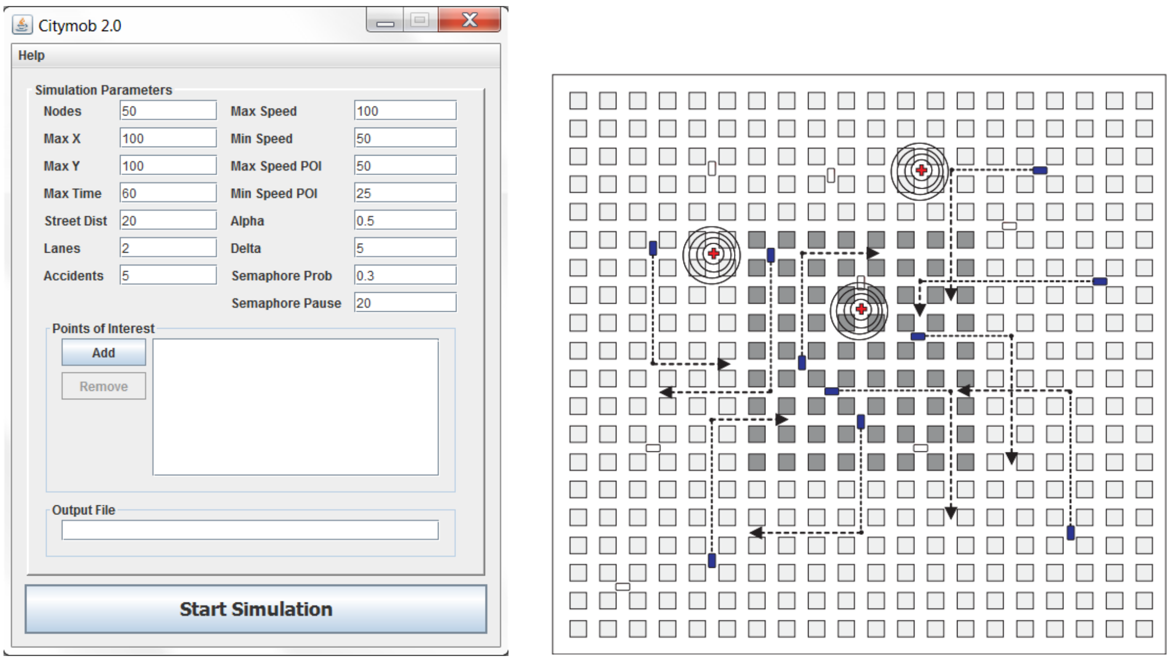


Figure 20. a) Citymob GUI b) Downtown Scenario

(Martinez, Toh, Cano, Calafate & Manzoni 2008)

Upon running a simulation scenario, the result is a log file consisting of two parts. The first part defines the initial (x, y) coordinates of each vehicle. The second part is a log for every car's speed and positions as well as stoppages due to semaphores. It also specifies which nodes had accidents; those nodes remain static for the rest of the simulation period.

Figure 21 shows a comparison of the main features supported by the VANET simulators discussed previously.

	VanetMobiSim	SUMO	MOVE	STRAW	FreeSim	CityMob
Software						
Portability	✓	✓	✓	✓	✓	✓
Freeware	✓	✓	✓	✓	✓	✓
Opensource	✓	✓	✓	✓	✓	✓
Console	×	✓	✓	—	×	✓
GUI	✓	✓	✓	✓	✓	✓
Available examples	✓	✓	✓	—	✓	×
Continuous development	×	✓	×	×	—	✓
Ease of setup	Moderate	Moderate	Easy	Moderate	Easy	Easy
Ease of use	Moderate	Hard	Moderate	Moderate	Easy	Easy
Maps						
Real	✓	✓	✓	✓	✓	×
User defined	✓	✓	✓	—	×	×
Random	✓	✓	✓	×	×	✓
Manhattan	×	×	×	×	×	✓
Voronoi	✓	×	×	×	×	×
Mobility						
Random waypoint	✓	✓	✓	×	×	✓
STRAW	×	✓	✓	✓	×	×
Manhattan	×	✓	✓	×	×	✓
Downtown	×	×	×	×	×	✓
Traffic models						
Macroscopic	×	×	×	×	✓	×
Microscopic	✓	✓	✓	✓	✓	✓
Multilane roads	✓	✓	✓	✓	—	✓
Lane changing	✓	✓	✓	✓	—	✓
Separate directional flows	✓	✓	✓	✓	—	✓
Speed constraints	✓	✓	✓	✓	✓	✓
Traffic signs	✓	✓	✓	✓	—	✓
Intersections management	✓	✓	✓	—	—	×
Overtaking criteria	✓	—	—	—	—	×
Large road networks	—	✓	✓	✓	—	✓
Collision free movement	—	✓	✓	—	—	✓
Different vehicle types	×	✓	✓	—	×	✓
Hierarchy of junction types	×	✓	✓	—	×	×
Route calculation	✓	✓	✓	✓	✓	×
Traces						
ns-2 trace support	✓	×	✓	×	×	✓
GloMoSim support	✓	×	✓	×	×	×
QualNet support	✓	×	✓	×	×	×
SWANS support	×	×	×	✓	×	×
XML-based trace support	✓	×	×	×	×	×
Import different formats	✓	✓	✓	×	×	×

Figure 21. Features of different VANET Simulators (Martinez, Toh, Cano, Calafate & Manzoni 2009).

5.2. Network Simulators

5.2.1. NS-2

NS-2 is an open-source discrete event network simulator which can simulate both wired and wireless networks. The program is written in C++. However, simulation parameters such as time, wireless network parameters and initial conditions need to be specified in TCL. Many mobility generators produce trace files which are compatible with ns-2 including MOVE, VanetMobiSim, and CityMob. The result of the ns-2 simulation process is two files. The first is an event trace file which contains a log of packets transmission, forwarding, drop and delivery. The second is an event animation file which can be used by a tool called *nam* to visualize the simulation process.

5.2.2. OMNeT++

"OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators" (Varga, András 2013). It runs on Windows, Mac OS X Linux, and other Unix-like systems. Using OMNeT++ Eclipse-based IDE and other tools, it is possible to develop model frameworks for a wide range of simulations including sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling and photonic networks. Modules are programmed in C++ then assembled into larger components and models using a high-level language (NED). The following frameworks are the most popular simulation frameworks for OMNeT++. (Varga, András 2013.)

The INET Framework: Supporting several models for internet (TCP, UDP, IPv4, IPv6, OSPF, BGP), wired and wireless link layer protocols (Ethernet, PPP, IEEE 802.11) as well as MANET protocols, INET is regarded as the standard protocol model library for OMNeT++ and it supported through regular updates and fixes by OMNeT++ team and community members.

INETMANET: Is a framework for simulating mobile ad-hoc networks. It is maintained and updated by Alfonso Ariza Quintana.

MiXiM: A wireless network simulator specialized for simulating the lower layers of networks with its rich library models for radio wave propagation, interference estimation, radio transceiver power consumption and wireless MAC protocols. MiXiM was created by merging several earlier OMNeT++ frameworks and it is planned to be merged into the INET framework in the future.

Castalia: Is a wireless sensor network simulator capable of running and visualizing realistic and large parametric simulations.

Veins: Veins is a vehicular ad-hoc network simulation framework. It utilizes SUMO for mobility generation and OMNeT++ for network simulation. Veins runs both simulators simultaneously and connects them via a TCP socket allowing bidirectionally-coupled simulations. It is able to simulate the IEEE 802.11p and IEEE 1609.4 DSRC/WAVE standards through adding their models to the OMNeT++ MiXiM Framework. OpenStreetMaps are extensively supported; it is possible to import whole scenarios including buildings, speed limits, lane counts, traffic lights and even access and turn restrictions. A Two-Ray Interference Model is available for simulating radio path loss as well as a simple obstacle model for simulating signal attenuation by buildings. **Figure 22** explains how Veins integrates SUMO and OMNET++ to allow VANET sinulations.

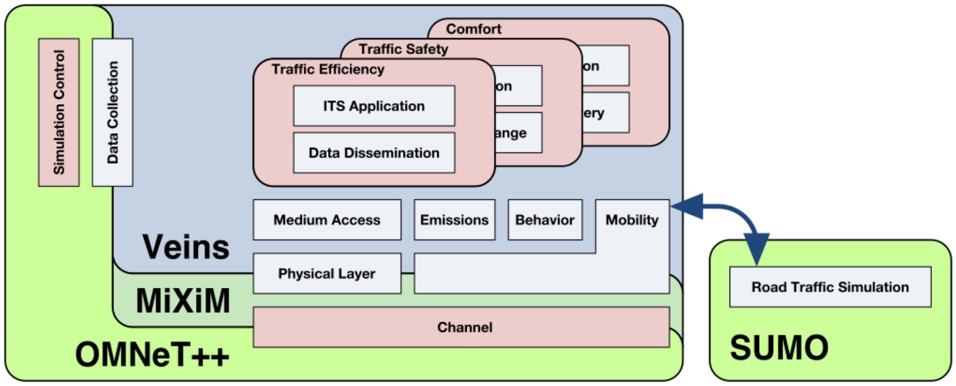


Figure 22. Veins (Sommer 2012)

5.2.3. Scalable Wireless Ad hoc Network Simulator (SWANS)

SWANS was developed as highly scalable wireless networks simulator. It offers faster runtime and less memory usage compared to ns-2 and GloMoSim. Node movement scenario and communication scenario are specified via a java input file. Ready-made applications can be selected for any node. Users are also able to build and execute their own custom made applications at the node's application layer. (Martinez, Toh, Cano, Calafate & Manzoni 2009).

In addition, many other powerful commercial network simulators are available such as OPNET and QualNet (the commercial version of GloMoSim) but those are not discussed here due to their high cost and the fact that their source code is not open.

5.3. VANET Simulators

5.3.1. Traffic and Network Simulation Environment (TraNS)

TraNS integrates SUMO and ns-2 to generate realistic VANET simulations. It takes into account the mutual influence between broadcasted messages and vehicles mobility; for example, cars slowing down in case of receiving a warning message about an accident which happened nearby. In the mobility simulation side, TraNS can generate a map using XML data or by loading a TIGER or Shapefile maps. Several routes can be loaded from XML data or defined manually. For each route, the user needs to specify the starting and ending point and the number of vehicles following that route. In the network simulation side, several parameters can be adjusted. They include the wireless channel type, radio propagation model, routing protocol, MAC type and even the antenna type.

TraNS has two distinct modes of operation. The network centric mode is used to evaluate VANET applications which do not influence driver's behavior such as music broadcasting and other comfort applications. The application centric mode is used to evaluate collision avoidance, emergency braking and other safety applications which have a direct influence on the driver's behavior. (Piorkowski, Raya, Lugo, Papadimitratos Grossglauser & Hubaux 2008).

The GUI of TraNS is shown in **Figure 23.**

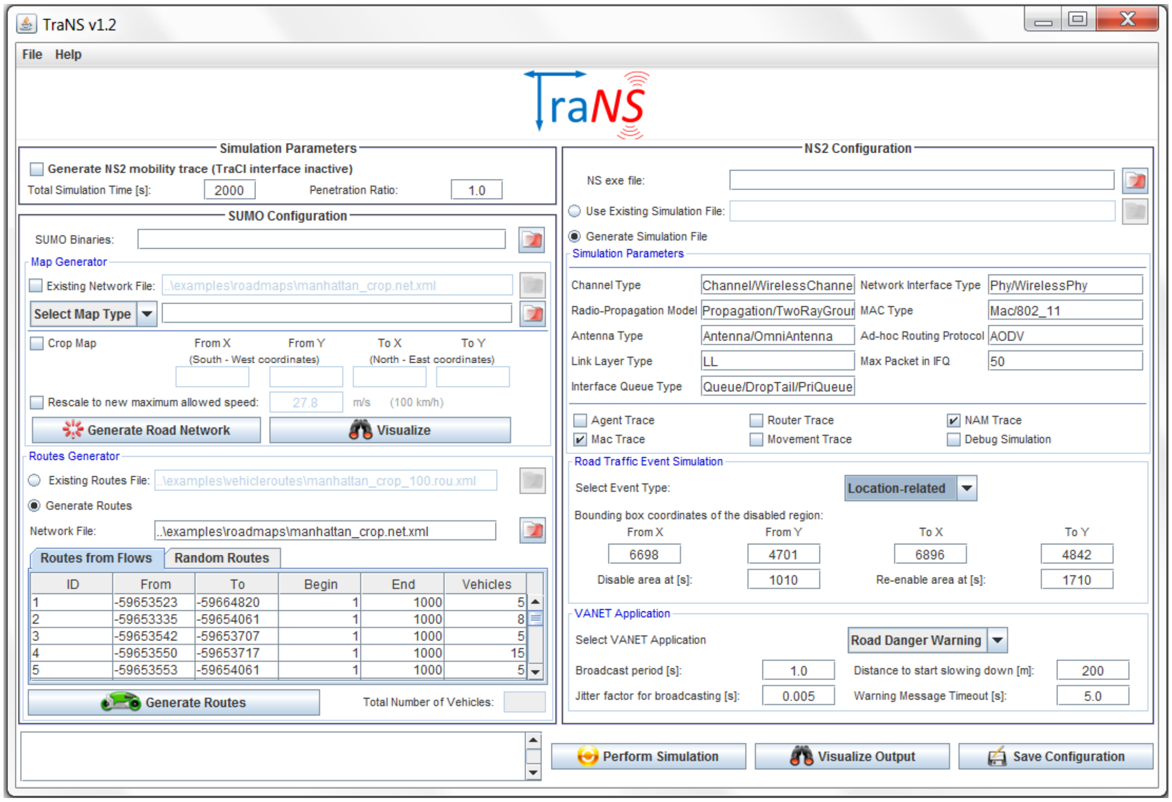


Figure 23. TraNS GUI.

5.3.2. MobiREAL

MobiRAEL is able to simulate mobility of both humans and vehicles as a probabilistic function using models written in C++. Vehicles change their speed and direction as a reaction to the environmental obstacles and their neighboring nodes as well as data received via applications. The simulator supports a collision avoidance algorithm for pedestrians and for modeling congestion of vehicles. It is also possible to simulate several mobility models simultaneously. (Martinez, Toh, Cano, Calafate & Manzoni 2009.)

5.3.3. National Chiao Tung University Network Simulator (NCTUns)

NCTUns Supports simulating a large variety of networks including Ethernet-based fixed Internet, IEEE 802.11b wireless LANs, IEEE 802.11e quality of Service wireless LANs, IEEE 802.16d WiMAX wireless networks, DVBRCS satellite networks, wireless vehicular networks for Intelligent Transportation Systems (including V2V and V2I), multi-interface mobile nodes for heterogeneous wireless networks, IEEE 802.16e mobile WiMAX networks and IEEE 802.11p/1609WAVE wireless vehicular networks. (Martinez, Toh, Cano, Calafate & Manzoni 2009.)

Maps can be easily created using different road segments such as single-lane roads, multilane roads, crossroads, T-shape roads, and lane-merging roads. It is also possible to import and crop Shapefile maps as shown in **Figure 24**. Obstacles with specific dimensions and signal attenuations can be added to represent buildings on the map. Those obstacles can be modified to obstruct nod's movement or driver's vision as well.

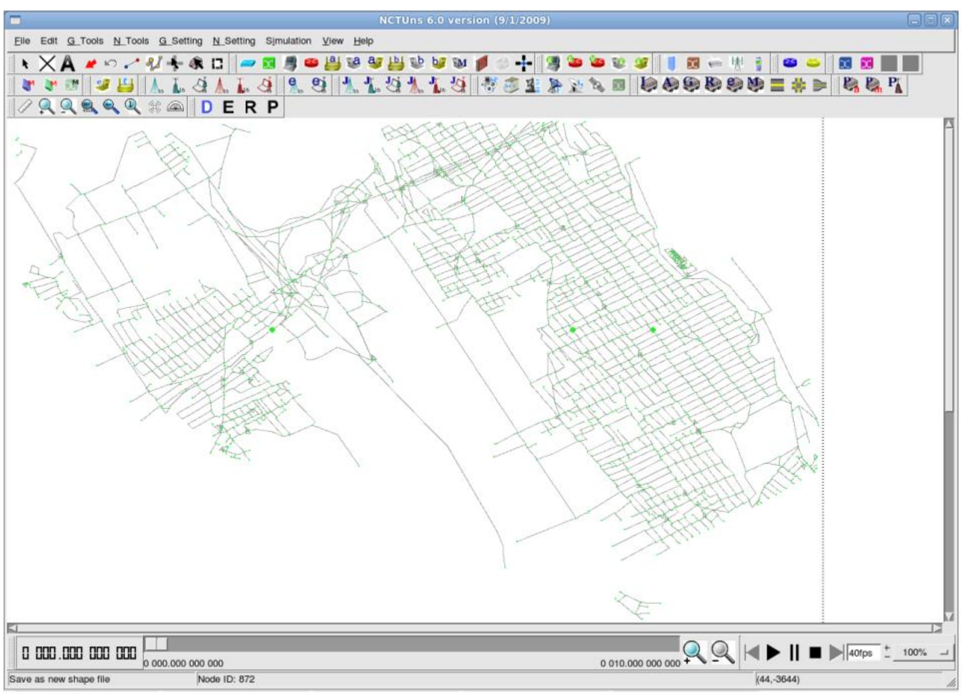


Figure 24. NCTUns GUI

Vehicles can be deployed automatically by defining their number and the average distance between two nodes or manually by clicking anywhere on the map. After deploying cars, they can be assigned by percentage to five different categories called car profiles. These profiles define the car's maximum speed, maximum acceleration and maximum deceleration. Users can also create and save their own profiles. Car's movement behavior is defined by the car agent (CA). It scans the neighbor nodes, surroundings and obstacles to decide the driver's behaviors. Five different car agents are available through .cc files, those are CarAgent (the default one), Down, Group, LaneSwitch, SlowDown and Broken. By using car agents it is possible to simulate overtaking, collision avoidance and car accidents among other VANET applications. Parameters of different network layers can be adjusted for each node as shown in **Figure 25.**

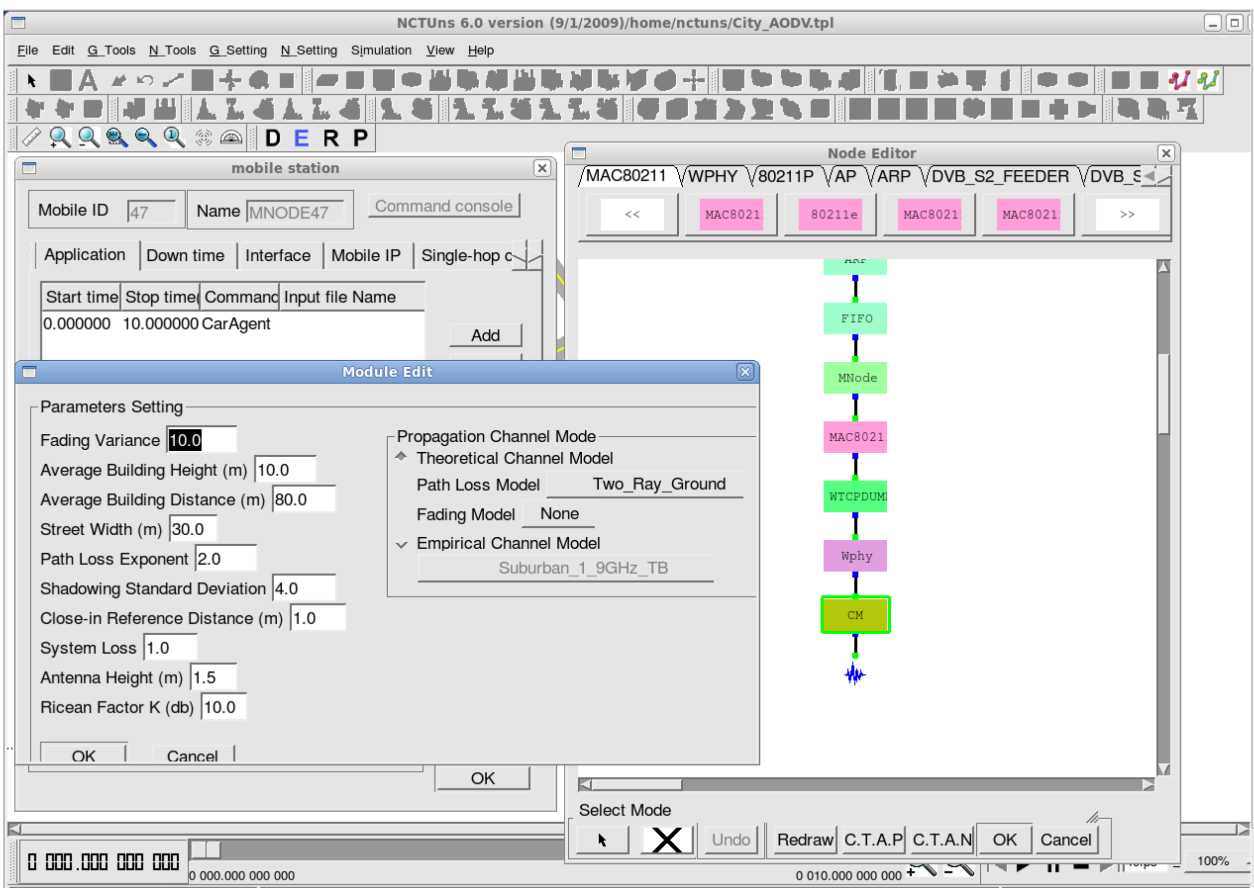


Figure 25. Editing VANET Node Properties in NCTUns.

NCTUns provides a highly integrated and professional GUI environment which makes it easy to draw network topologies, configure the protocol modules of each node (up from the application layer and down to the antenna type, transmission power and range), specify the moving, paths of mobile nodes, plot network performance graphs directly from the generated log files, and play back the animation of a logged packet transfer trace.

The main advantage of NCTUns is that “its network protocol stacks includes the Linux kernel protocol stack, including TCP/IP and UDP/IP, and the user level protocol stack and the MAC and PHY layer protocols”(Martinez, Toh, Cano, Calafate & Manzoni 2009). This allows running any Linux compatible application on any of the simulated nodes. The main disadvantage of NCTUns is the integration of cars mobility logic with the network simulation code which makes it complicated to modify and add new features to the simulator. Also, because it is based on the Linux kernel, NCTUns compatibility is limited to Fedora Linux (Fedora 12 for NCTUns 6.0) which is a reason for some researchers to avoid using it.

Figure 26 illustrates NCTUns architecture and VANET simulation process.

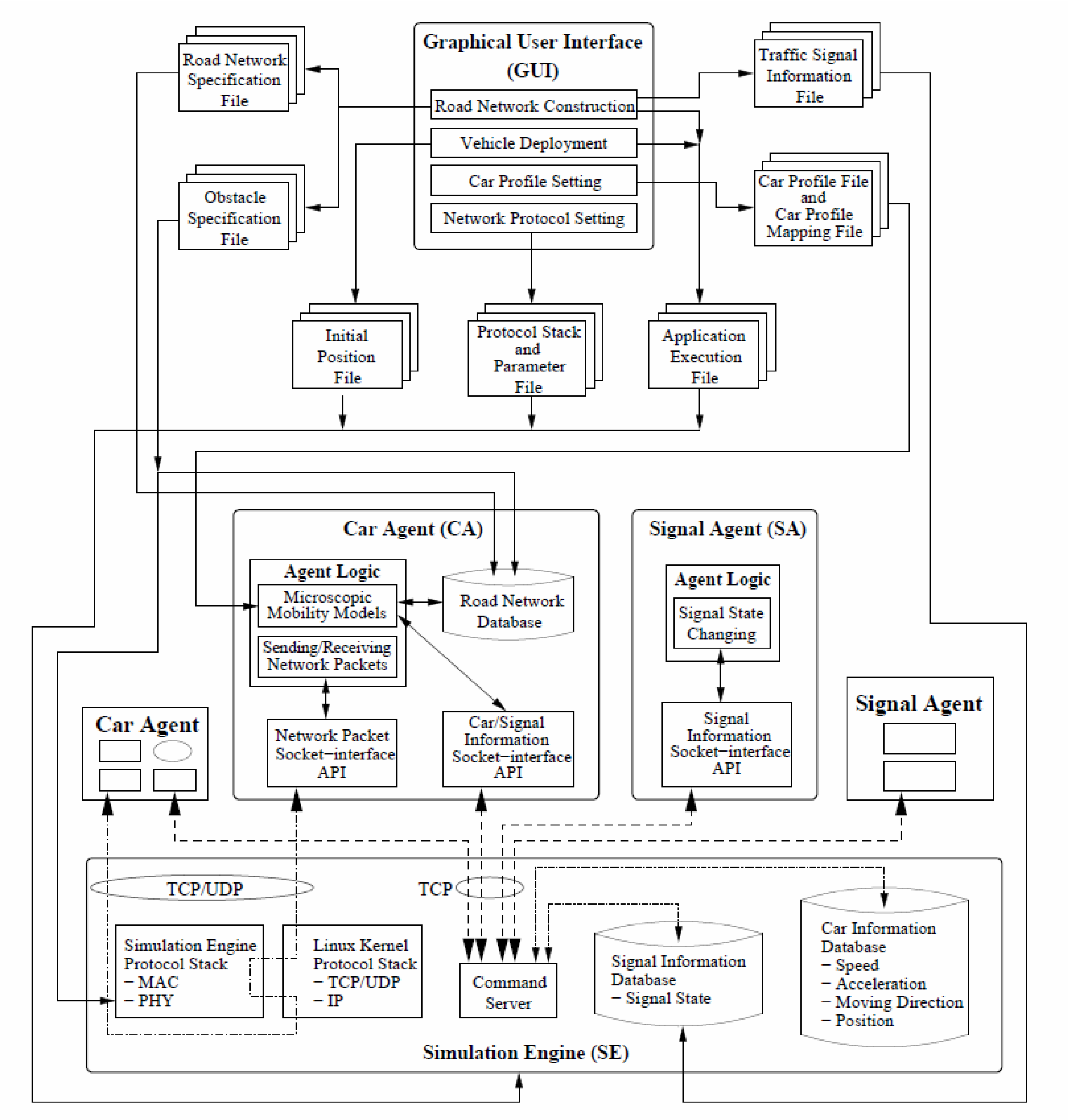


Figure 26. NCTUns Architecture (Wang & Chou).

6. SIMULATIONS

The first stage of simulation is designing a map for the mobility model simulation process. For the simulations carried out in this thesis, some use real city maps and some use manually created maps to simulate a specific scenario. To design a manual map using MOVE, the user needs to define nodes and edges as shown in **Figure 27** and **Figure 28** respectively. A node represents an intersection or traffic light containing an ID and location coordinates where the coordinates are defined in meters. Edges are the streets which connect nodes. By specifying from which node to which node the edge is placed, direction of cars movements are defined. Additionally, number of lanes and maximum allowed speed for vehicles must be defined.

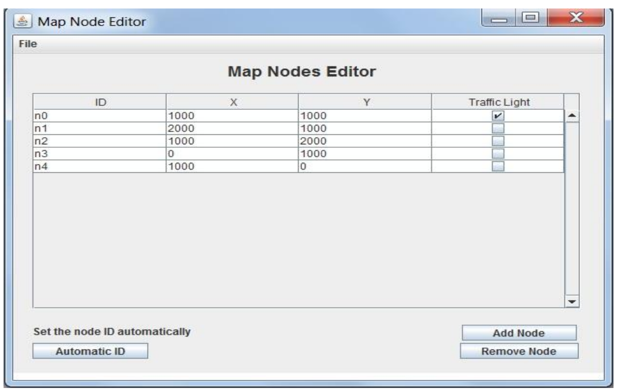


Figure 27. MOVE's Map Nodes Editor

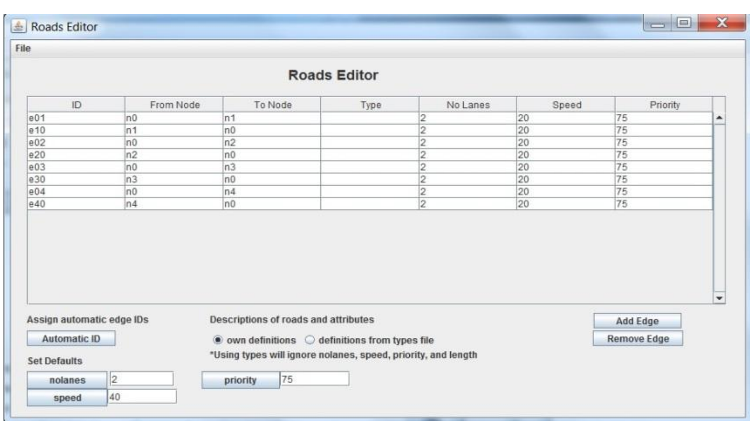


Figure 28. MOVE's Roads Editor

After creating and saving the nodes and edges files, the configuration editor tool links those two files and produces a configuration file which can be used by the map generator tool to create a map. As for real city maps, they were obtained by using VanetMobiSim to browse and parse "http://www.openstreetmap.org" as shown in **Figure 29.**

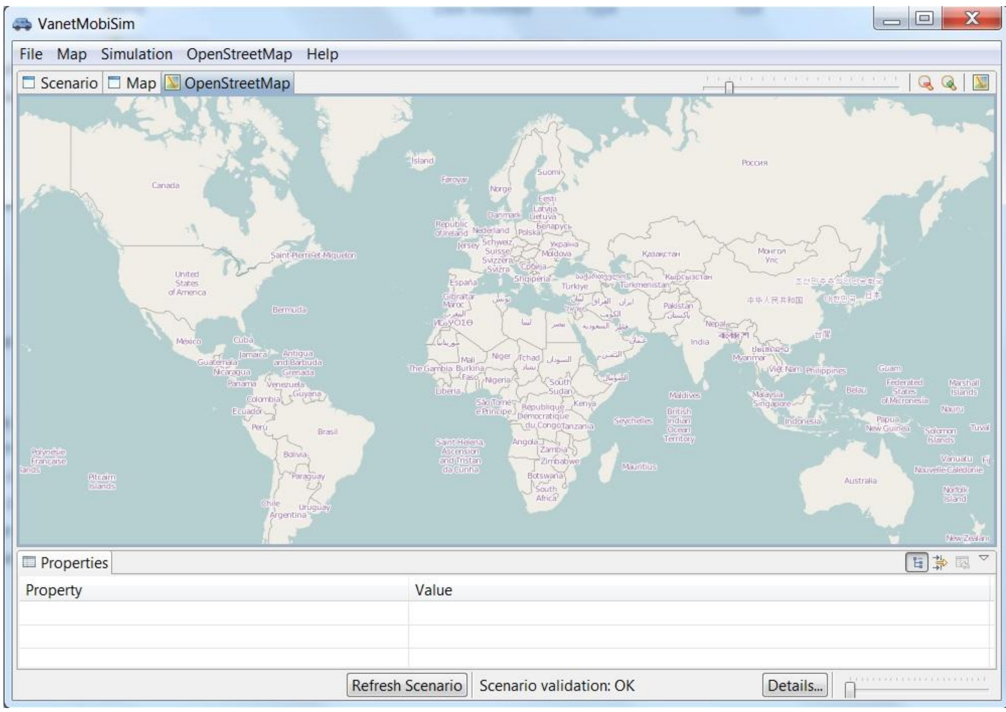


Figure 29. VanetMobiSim's GUI

The saved files are in .osm format. Netconvert is a powerful tool which comes with sumo windows binaries and can be used to convert several map file formats including .osm into a compatible SUMO map file.

The second stage is editing vehicle's movement. In this thesis, a combination of automatically generated car following flows and manually defined cars are used. For every flow the ID of source and destination edges need to be specified. Cars will travel from the beginning node of the source edge towards the beginning node of the destination edge. The path of the trip is calculated based on shortest path. Cars obey map parameters such as maximum allowed speed and traffic lights and cannot move in a wrong way. Overtaking is possible in case of multilane streets. For every flow a start time, end time and a number of vehicles are defined. The specified number of vehicles will be generated during the time period at equal intervals one vehicle at a time. The vehicle flow definition editor is shown in **Figure 30**.

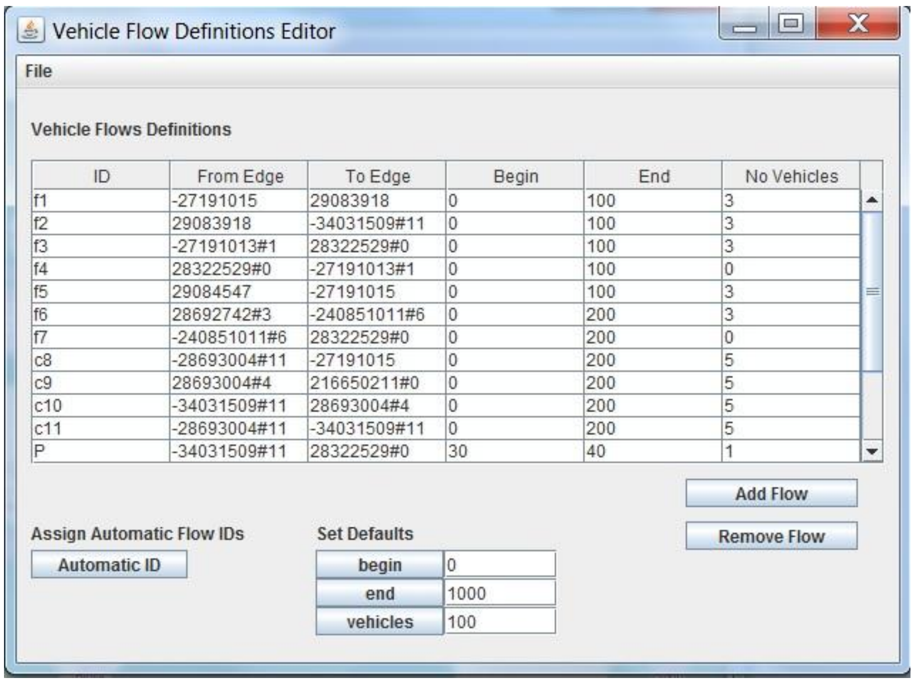


Figure 30. Editing Vehicular Flows with MOVE.

The Manual Vehicle Route Editor is a powerful tool which allows more control over the cars behavior. Unique types of vehicles can be created and named. Car's acceleration, deceleration, maximum speed and length can be defined as shown in **Figure 31**.

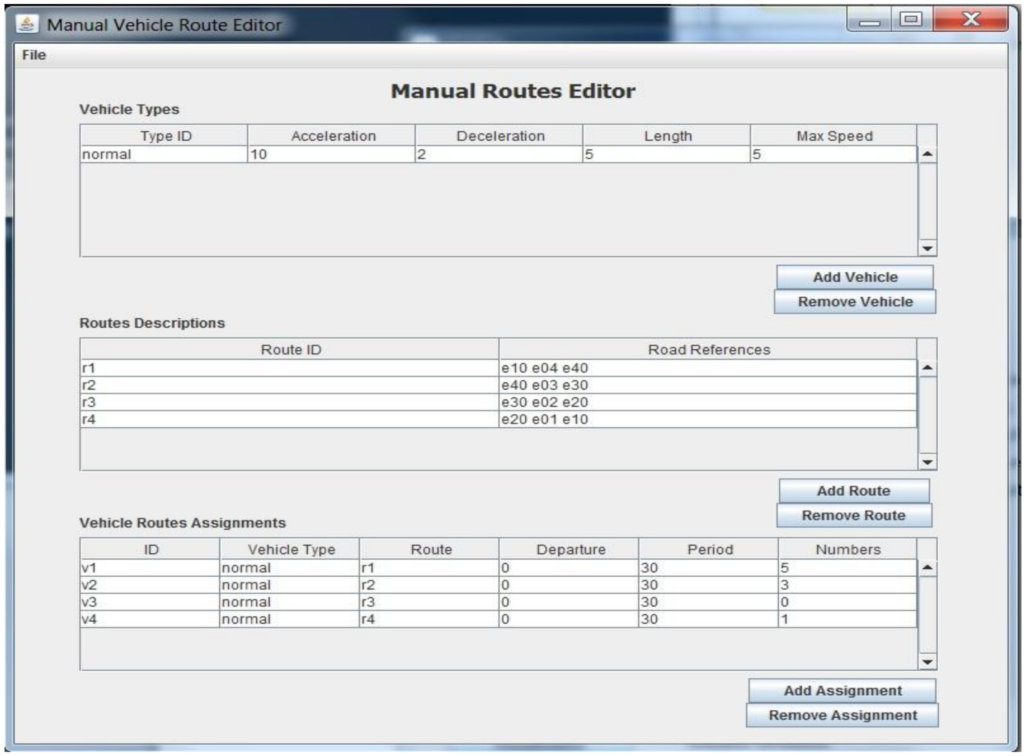


Figure 31. Editing Routes Manually with MOVE.

In the route descriptions, the user can explicitly state which edges every route will traverse between source and destination. To produce the desired vehicular flow, a vehicle type needs to be coupled with a route using the vehicle routes assignments tool.

The last stage is mobility simulation. A configuration file linking the map with the routes file is produced. This file can be visualized by SOMO's visualization tool to carefully inspect the resulting mobility model. After that the actual simulation can be executed by SUMO to produce a mobility trace file. The mobility trace is a log file defining X, Y coordinates for every node at given time intervals. This file can be imported along with its corresponding map into MOVE's ns-2 script generator which is shown in **Figure 32**. The ns-2 script generator is a GUI which allows easy manipulation of key parameters concerning VANET simulations in ns-2 including used routing protocol, antenna type, MAC type, and radio propagation model.

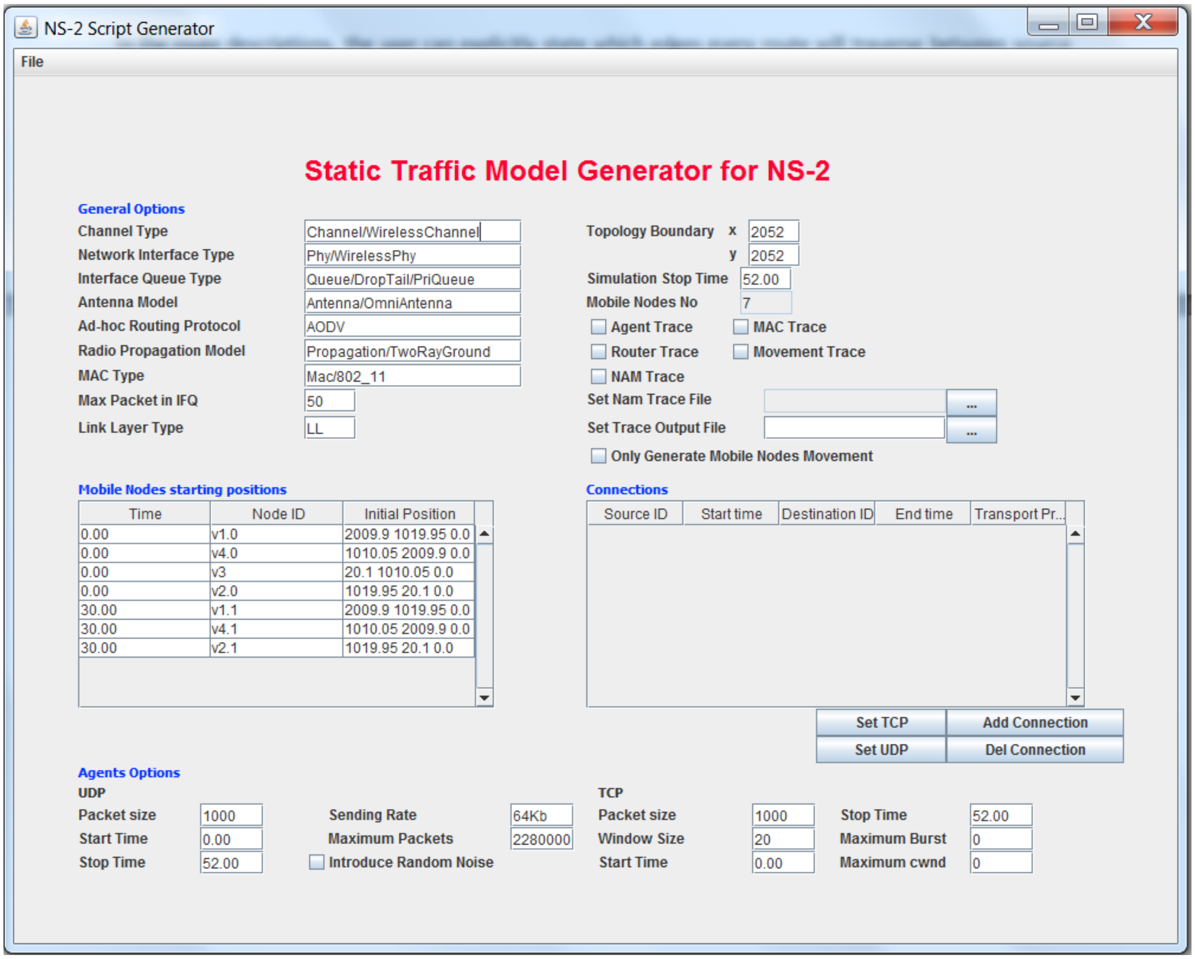


Figure 32. Generating ns-2 Scripts with MOVE.

Using the script generator, TCP or UDP packets can be transmitted via wireless connections between any two of the simulated nodes in the mobility model. The script generator produces a Tool Command Language (TCL) file directly executable by ns-2. **Figure 33** illustrates the steps of MOVE simulation process.

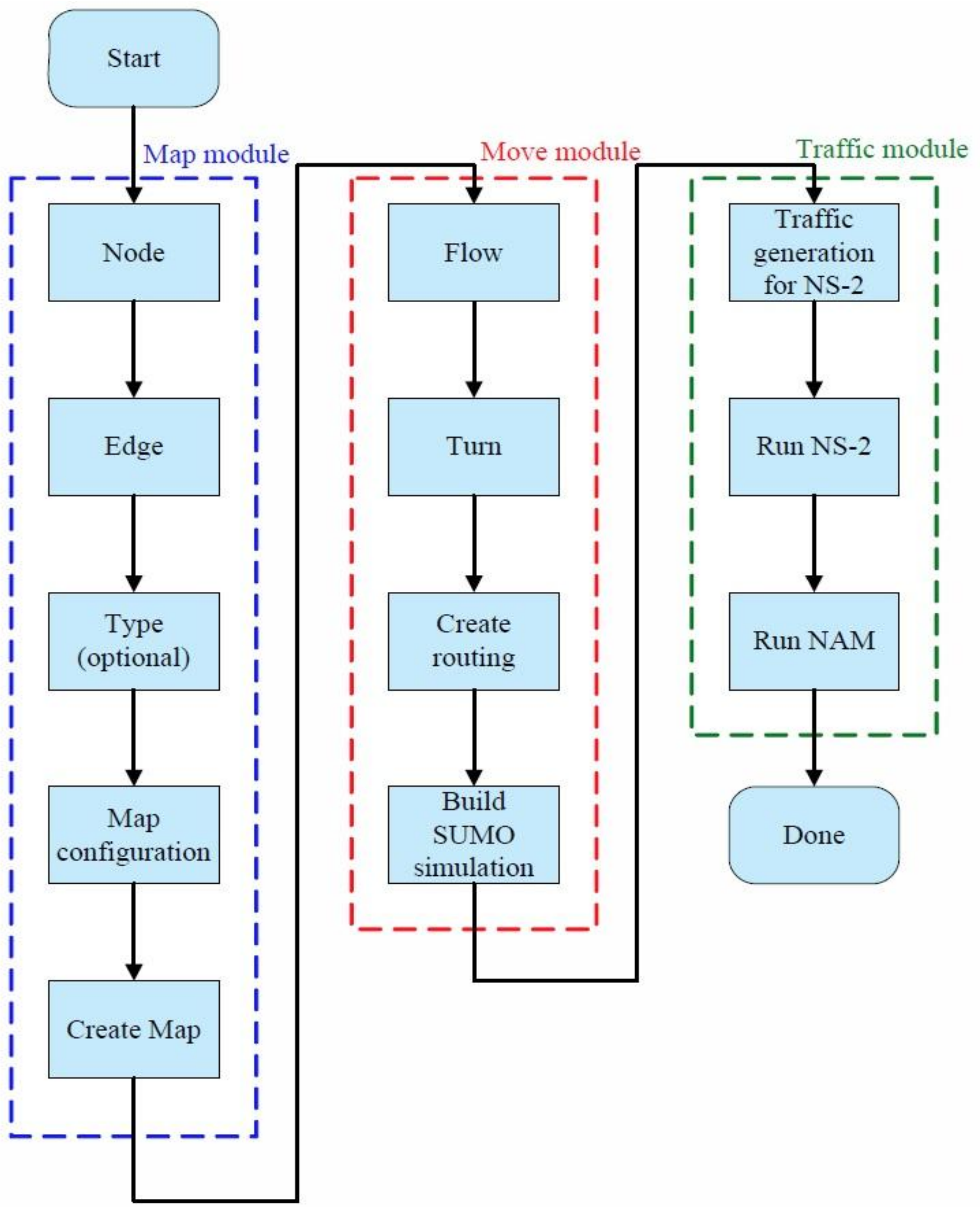


Figure 33. Simulation Process Using MOVE (Chou 2011).

For compatibility reasons, ns-2 was installed on a virtual machine running Ubuntu 12.04 LTS. Natively, ns-2 supports several ad-hoc routing protocols including AODV, DSR and DSDV. UM-OLSR was downloaded from its author's (Francisco J. Ros) website and patched into the installed ns-2. Upon running a simulation, ns-2 produces two trace files. The first one is a .nam file which can be used to animate packets transmission and nodes movement process with the ns-2 visualization tool as shown in **Figure 34**. This is useful to get a rough idea about the simulation results.

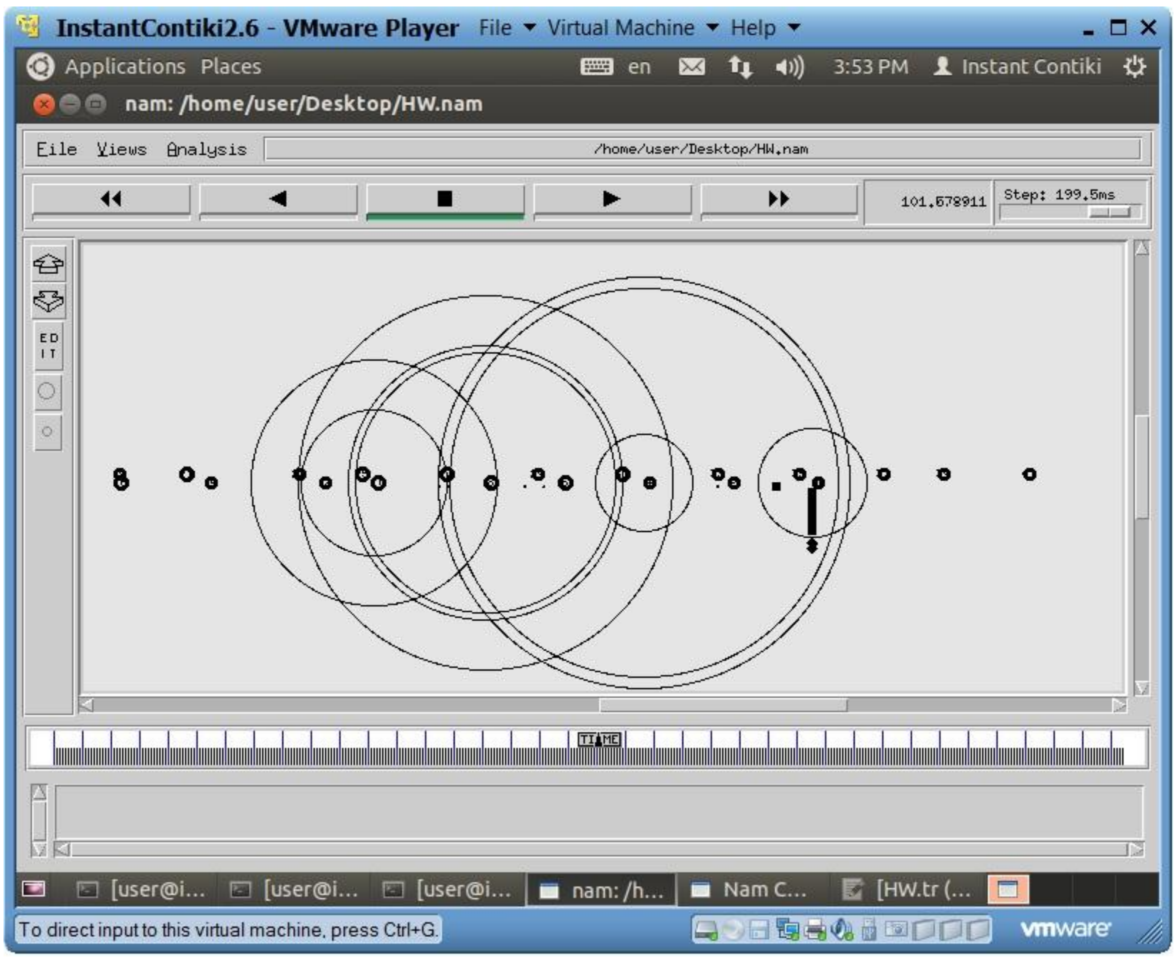


Figure 34. VANET Simulation Visualization Using nam.

The second file is a trace file which is basically a log for transmitted packets. It logs the time for every transmitted, received, and dropped packet at every node. The trace file can be either in old or new trace format. In this thesis, the new trace format was used because it offers better compatibility with the used trace analyzing software. A detailed description of the new ns-2 trace file format can be found in the following link “http://www.isi.edu/nsnam/ns/doc/node186.html”. This trace file is converted to simpler and faster format using a tool called *trconvert*. The resulting file has reduced loading time and crashing probability when loaded with the trace file analyzer *trace graph*. Trace graph GUI is shown in **Figure 35**.

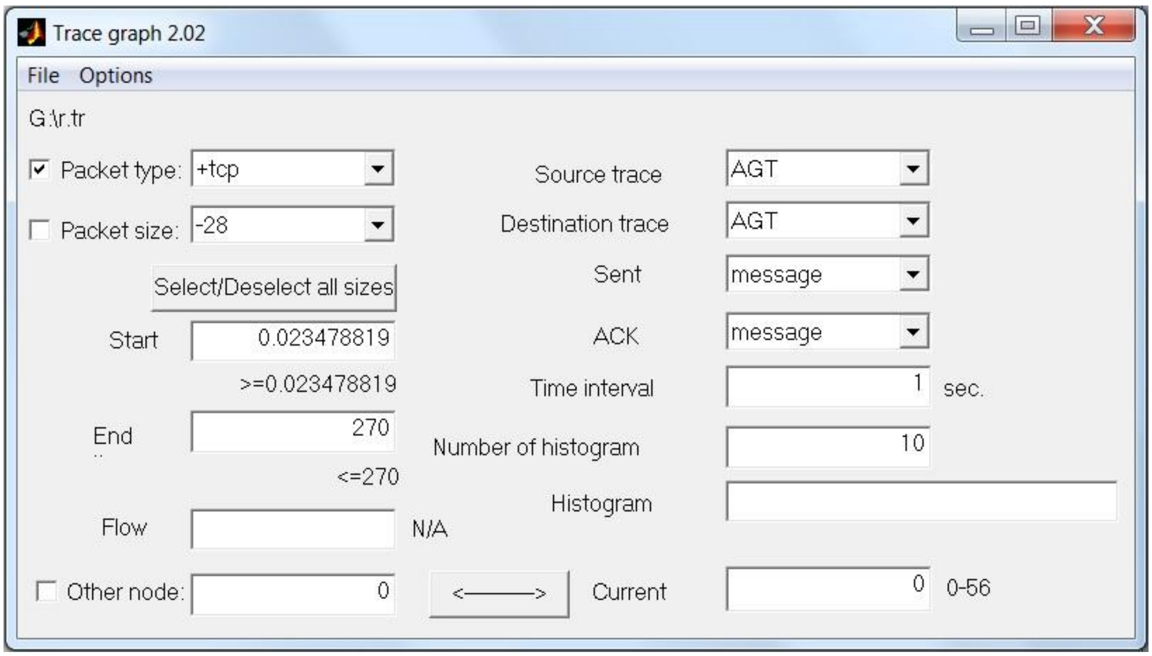


Figure 35. Loading ns-2 Trace File in Trace Graph.

Using trace graph, it is possible to set useful filters. For example, selecting which node to examine and limiting the time range in which data are calculated and graphs are plotted. One other important function is the ability to filter packet types; for TCP connections only TCP packets are selected while other packets such as Address Resolution Protocol (ARP), Request to Send frame (RTS), Clear To Send frame (CTS), and Acknowledgment (ACK) packets are filtered out in order to give correct results regarding throughput calculations. In case of UDP, only Constant Bit Rate (CBR) packets are selected.

6.1. Scenario 1

The first simulation scenario is designed to test routing protocols performance with multi-hop communications. The map is a simple highway of multilane and 10Km length. Two car flows are generated; one running at a lower a speed of 28 m/s and the other at 33 m/s. The first car in the fast lane transmits to the first car in the slow lane (both in blue color). As cars in the faster lane keep overtaking cars in the slower lane, the number of intermediate hopes keeps increasing. This is reflected in an increased end to end delay leading to a decreased throughput. **Figure 36** is the visualization of scenario 1 displayed using SUMO's GUI.

Figure 36. Simulation Scenario 1.

Simulation setup:

Transmission time: 240 seconds.

Number of nodes: 18

MAC type: Mac/802_11

Ad-hoc routing protocols: AODV, DSDV and OLSR.

Radio Propagation Model: Free Space.

Transport Protocols: TCP and UDP

Results:

Output peaks at 0.7 Mb/sec in the beginning of the scenario because the sending and receiving nodes are directly connected without intermediate nodes. As the number of hops increases, throughput decreases in the three of the tested routing protocols. DSDV loses connection at the 80^{th} second indicating that it cannot operate with a large number of intermediate hops. OLSR and AODV maintain connection throughout the whole transmission time with approximately the same throughput. However, it is worth noting that AODV has a more stable connection. When connection is lost, OLSR takes few seconds before successfully re-establishing connection resulting in some zero received bits gaps as shown in **Figure 37**.

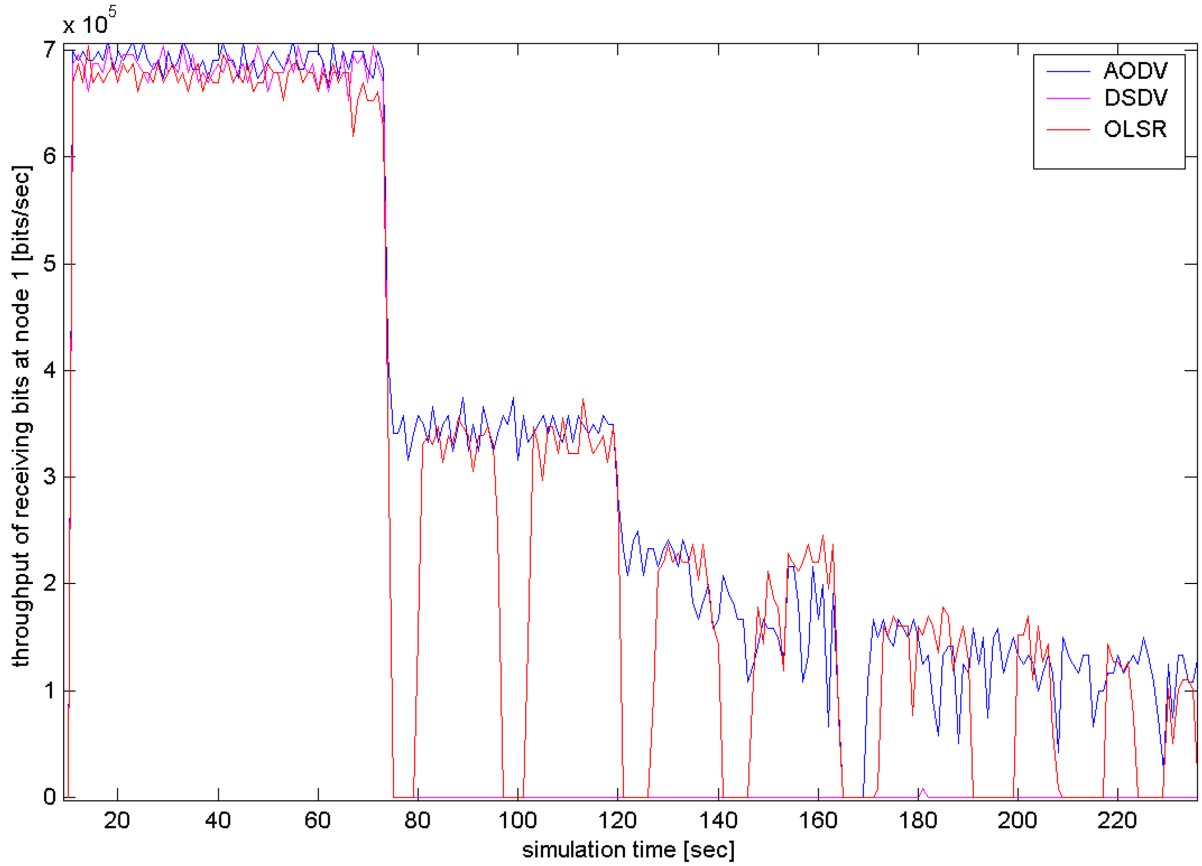


Figure 37. TCP Throughput for Scenario 1.

Figure 38 shows a comparison of the cumulative distribution functions of the delays.

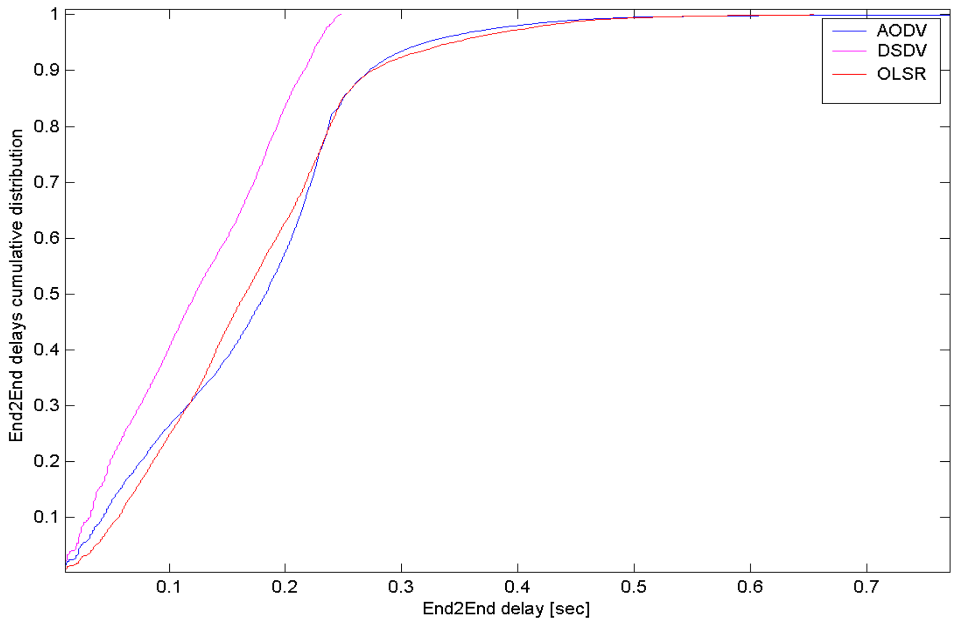


Figure 38. TCP Delay for Scenario 1.

DSDV has the lowest delays. OLSR and AODV have a similar CDF for delay. When sending packets with OLSR or AODV, over 50% of the packets are expected to arrive within 0.2 seconds and 90% of the packets will be received with less than 0.3 seconds of delay.

Table 3 shows the minimum, maximum and average end to end delays in seconds as well as the median and standard deviation (σ) calculated from the previous scenario for every routing protocol.

Table 3. TCP Packets Delay in Scenario 1.

	Minimum E2E Delay	Maximum E2E Delay	Average E2E Delay	σ	Median
AODV	0.0050919920	0.6537681730	0.1716732224	0.28866014	0.50010847
OLSR	0.0098262930	6.3630483810	0.1767772492	0.28858440	0.50025779
DSDV	0.0045719920	0.2477734980	0.1231930545	0.28865030	0.50019508

Using UDP packets the three routing protocols manage to maintain a connection for a higher percentage of the simulation time. AODV has slightly lower throughput when the number of intermediate nodes is high as shown in **Figure 39**.

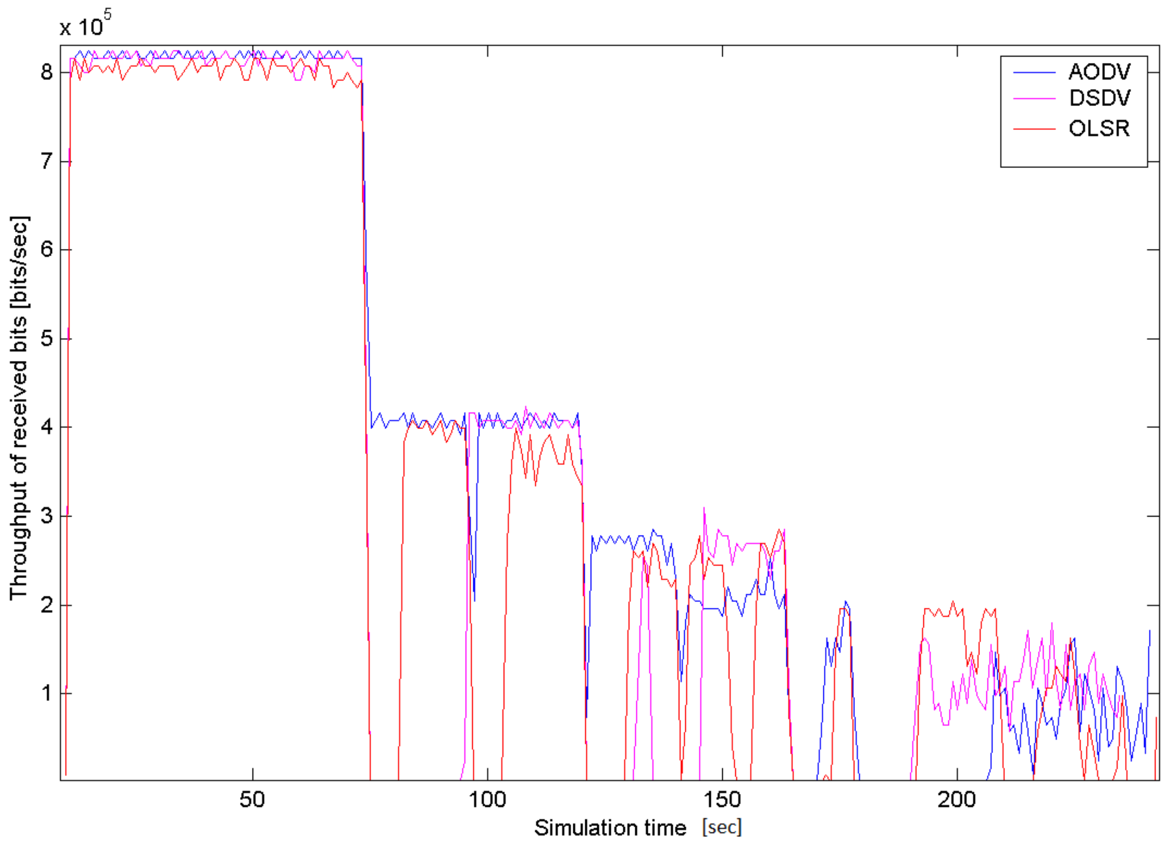


Figure 39. UDP Throughput for Scenario 1.

According to **Figure 40**, 70% of OLSR and DSDV packets have a delay of less than or equal to 0.5 seconds versus 63% of AODV packets. 90% of packets are expected to arrive with less than 2 seconds delay when using AODV or OLSR versus 85% for DSDV.

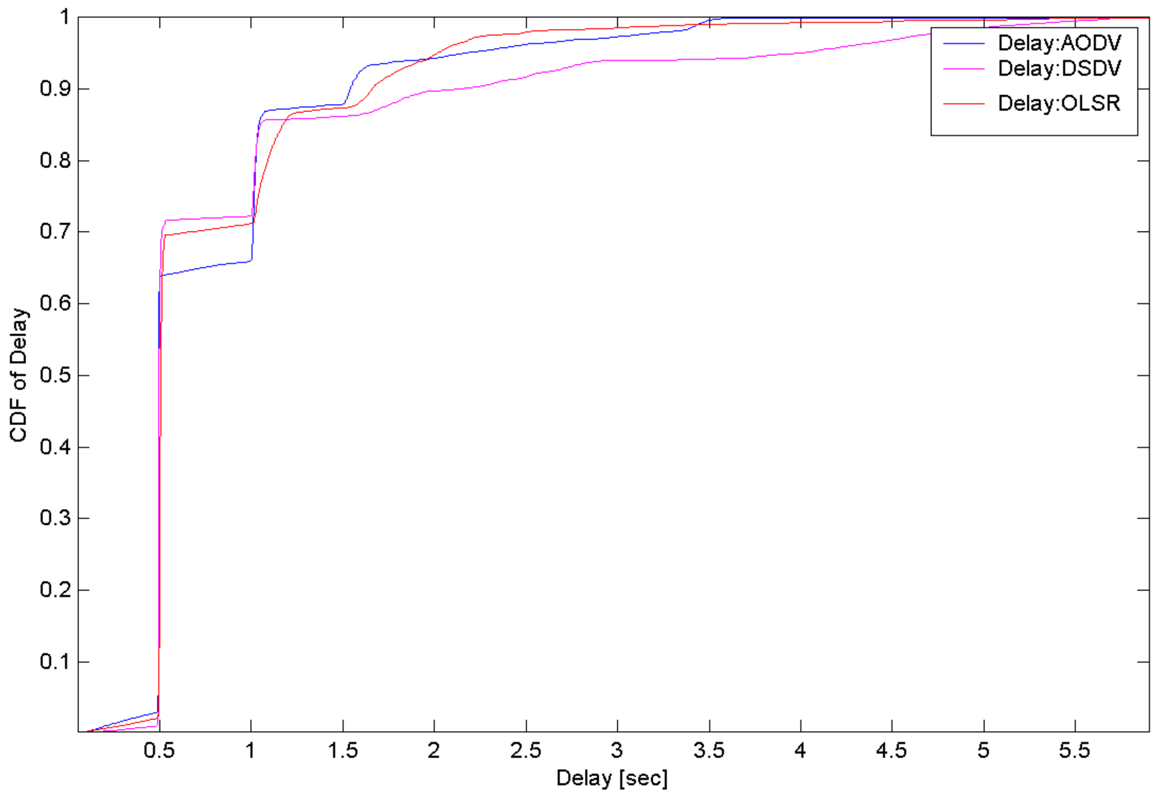


Figure 40. UDP Delay for Scenario 1.

Table 4. shows the minimum, average and maximum expected delays for AODV, OLSR and DSDV in seconds as well as their medians and standard deviations when using UDP packets.

Table 4. UDP Packets Delay in Scenario 1

	Minimum E2E Delay	Maximum E2E Delay	Average E2E Delay	σ	Median
AODV	0.0278405180	47.1854971770	0.8358558674	0.96577239	0.4972212000
OLSR	0.0125719920	13.1781788030	0.8198667509	0.70803833	0.5071783140
DSDV	0.0126519920	5.9846685140	0.9500924252	1.04989802	0.4983542355

The average delays are considerably higher in UDP than in TCP. This corresponds to results obtained by several other researchers when comparing delay between TCP and UDP packets. According to simulations carried out by (Gangurde, Waware & Sarwade 2012:1247), the average delay of received TCP packets was 0.787624 seconds while it was 1.930832 for UDP packets. (Giannoulis, Antonopoulos, Topalis, Athanasopoulos, Prayati & Koubias 2006) performed simulations to compare the quality of service and performance of TCP and UDP in multimedia applications over wireless networks. Their results show that UDP had higher mean for delay in all of the different simulations. One explanation for the delay problem in UDP is that in TCP, there is a feedback of acknowledgment packets allowing the protocol to reduce the transmission rate in case packets fail to reach their destination. In UDP however, there is no mechanism to inform the sender when connection is lost to the destination. As a result, the sender keeps generating and sending packets at the same rate. Those packets then cause the buffers of intermediate nodes to fill up resulting in long delays.

Table 5 shows a comparison of packet delivery ratio between the three inspected routing protocols. The packet delivery ratio is the percentage of sent packets which are not dropped neither lost but successfully received at their intended destination.

$$\text{Packet Delivery Ratio} = \frac{\text{Total number of sent packets at source node}}{\text{Total number of received packets at destination node}} \quad (22)$$

Table 5. Packet Delivery Ratio of UDP Packets in Scenario 1

	No. of sent packets	No of received packets	Packet Delivery ratio %
AODV	12990	10356	80%
OLSR	12495	9202	74%
DSDV	13333	8887	67%

AODV outperforms OLSR and DSDV in both packet delivery ratio as well as the total number of packets successfully delivered.

Figure 41 summarizes the results obtained so far.

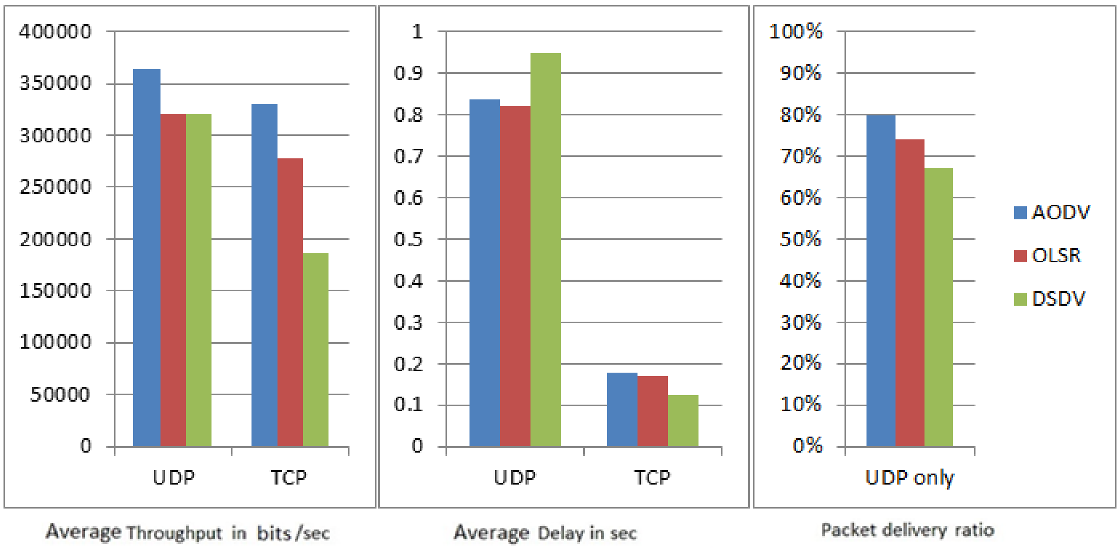


Figure 41. Summary of Scenario 1 Results.

AODV has the highest packet delivery ratio and average throughput. OLSR comes in the middle. DSDV offers the lowest delay in case of TCP packets but it has poor packet delivery ratio and throughput.

Next, the speed difference between the two lanes is increased to test the performance of routing protocols when changes in topology occur faster. In the faster lane cars run at 55 m/s while they run at 28 m/s in the slower lane. Simulation time is reduced to 110 seconds. First the results for TCP packets are shown in **Figure 42** and **Figure 43**.

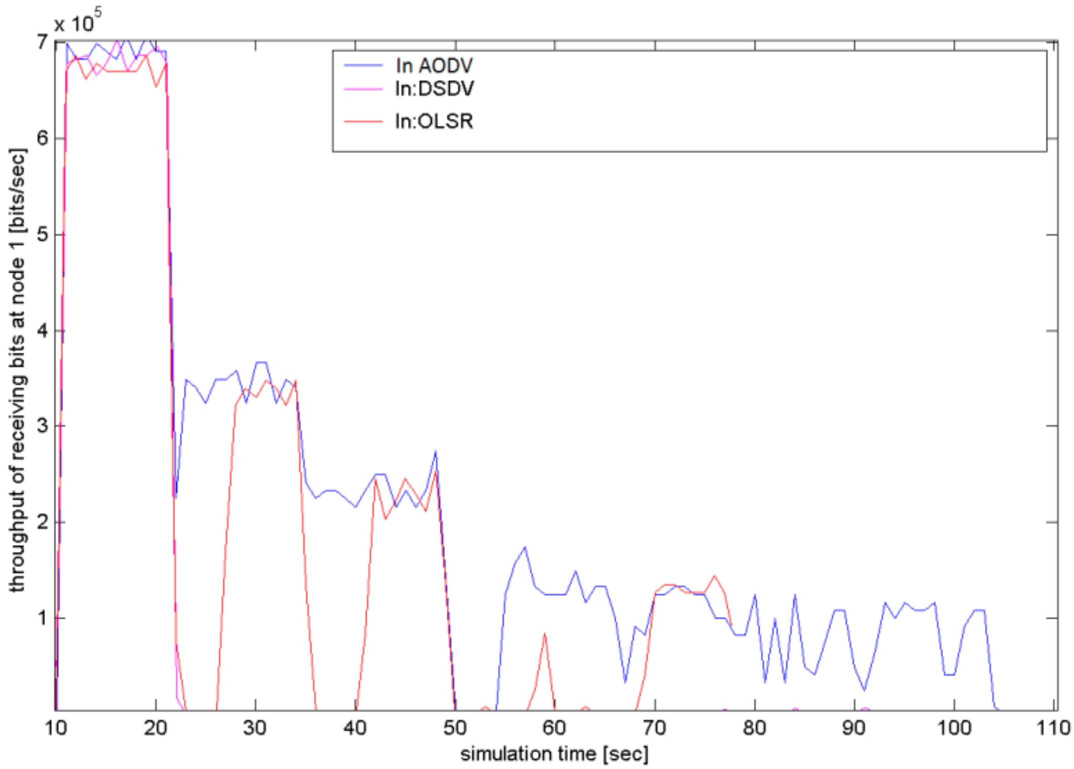


Figure 42. TCP Throughput When increasing Speed in Scenario 1.

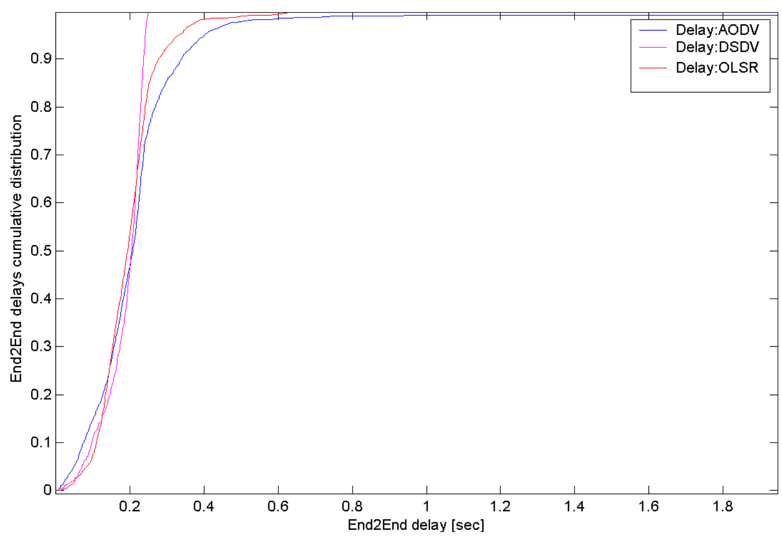


Figure 43. TCP Delay When increasing Speed in Scenario 1

Next, Results for UDP packets are shown.

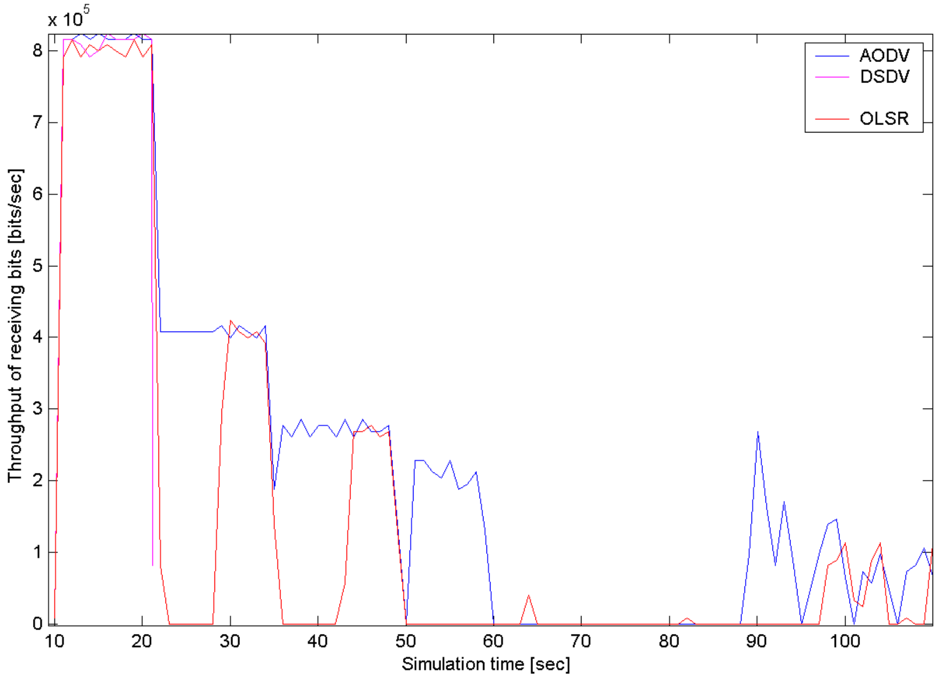


Figure 44. UDP Throughput When increasing Speed in Scenario 1

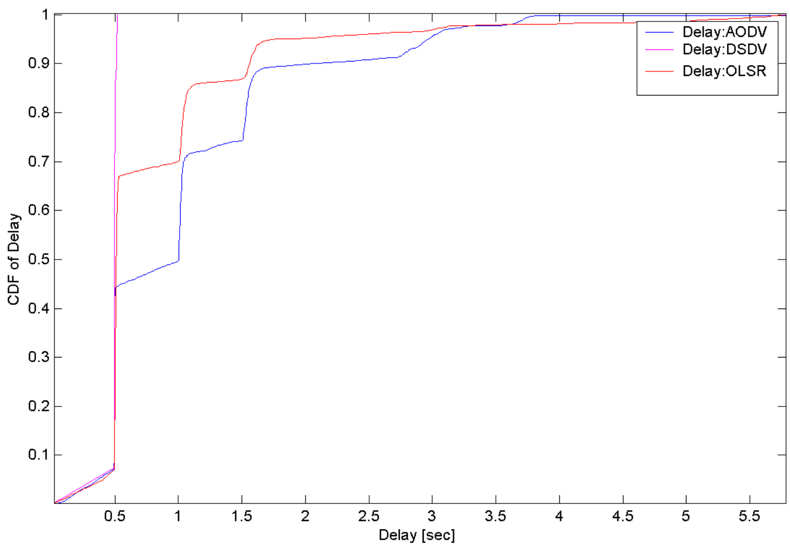


Figure 45. UDP Delay When increasing Speed in Scenario 1

As expected, with higher mobility, throughput and packet delivery ratio has decreased.

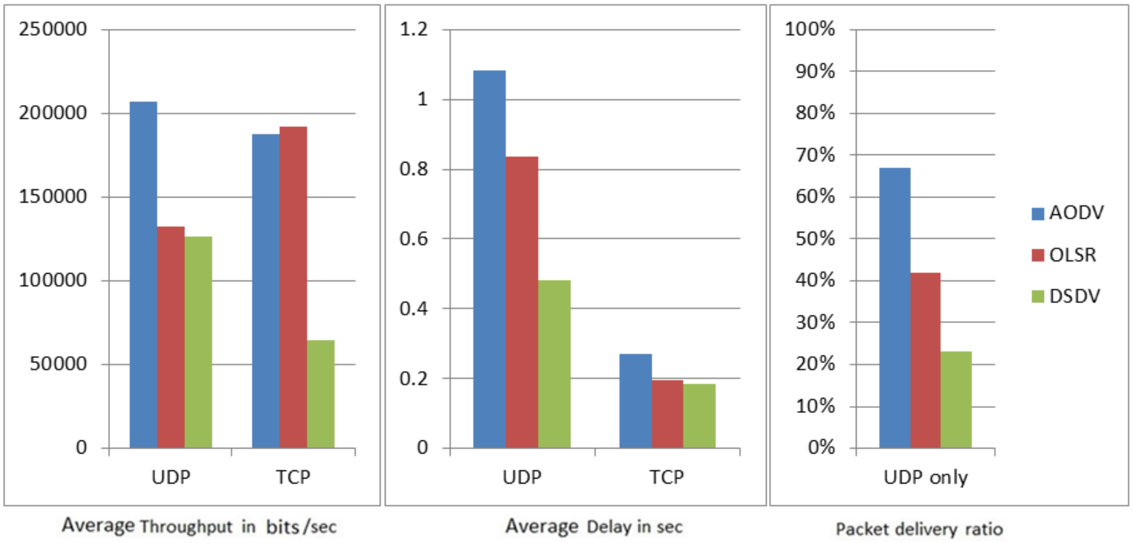


Figure 46. Summary of Results When Increasing Speed in Scenario 1.

AODV is be the most robust routing protocol with a reduction from 80% to 67% in packet delivery ratio while OLSR's was reduced by 32% and DSDV's was reduced approximately to one third from 67% to 23%. DSDV offers lowest delays in all of the carried simulations.

6.2. Scenario 2

This scenario is designed to inspect delay thouroughly. The map consistes of four streets of 1 Km length each connected at an intersection with a traffic light as shown in **Figure 47**.

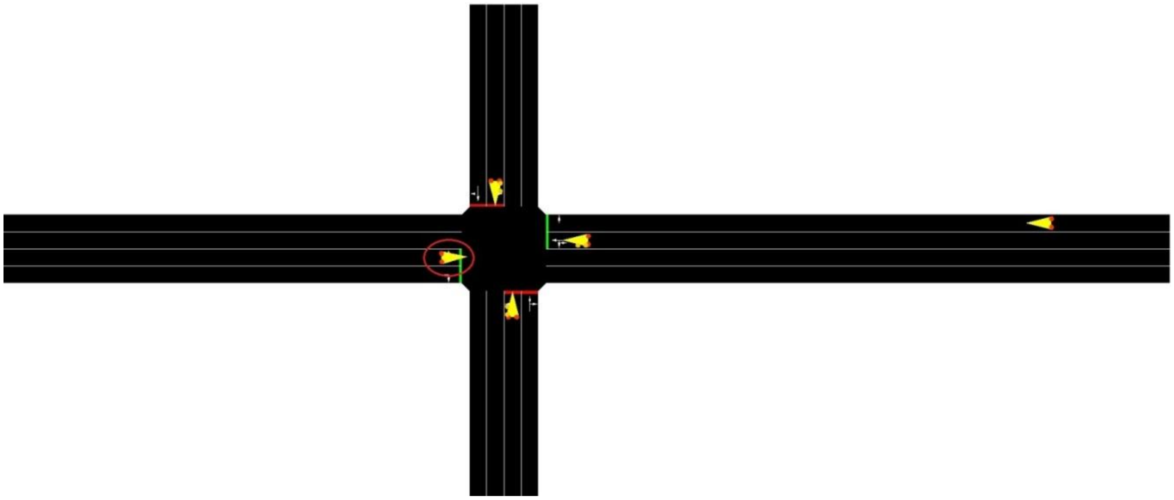


Figure 47. Scenario 2.

As the car inside the red circle prepares to take a turn it needs to warn the other cars. Two cars are coming from the street to its left, four cars from the street to its right and six cars from the street in front. That means three different numbers of hops in every direction. Average delays of packets reaching the last car in all of the three directions are plotted and compared for the three routing protocols as well as their CDF. Only TCP is examined since this scenario represents a safety application.

Simulation setup

Transmission time: 24 seconds.

Number of nodes: 13

MAC type: Mac/802_11

Ad-hoc routing protocol: AODV, OLSR and DSR

Radio Propagation Model: Two Ray Ground

Transport Protocol: TCP

Results:

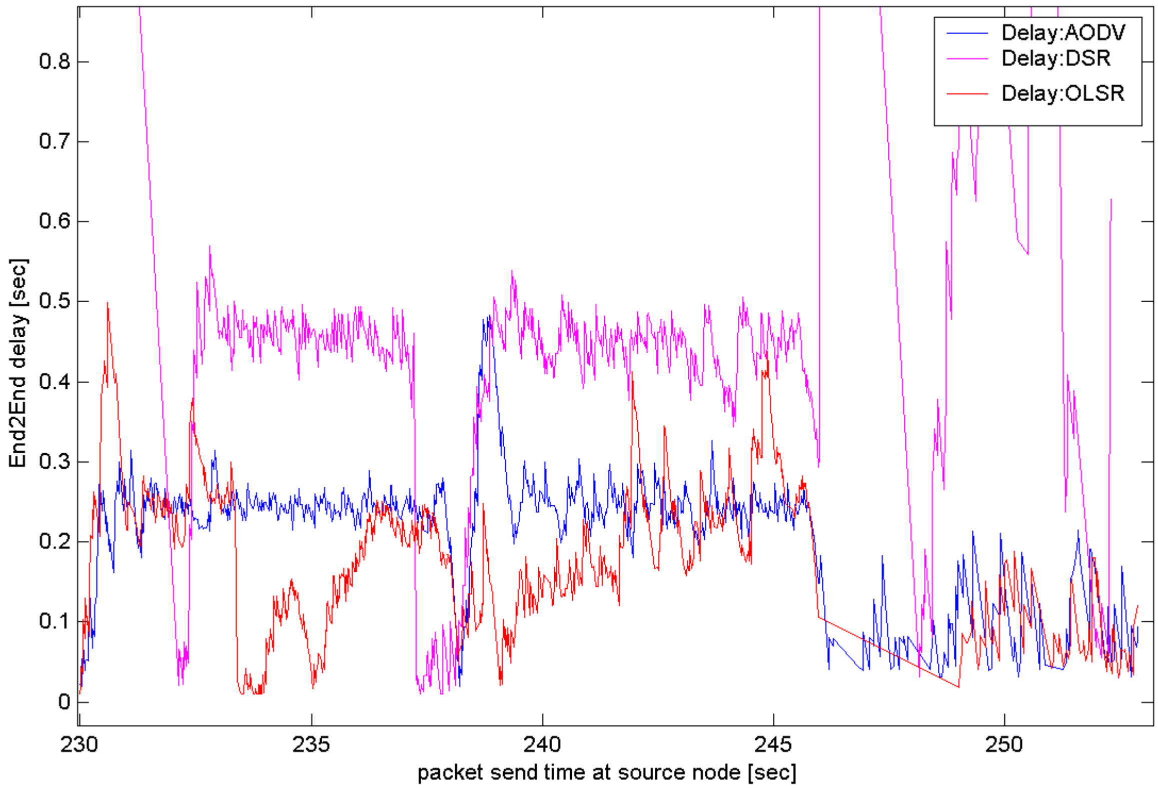


Figure 48. Throughput of Scenario 2.

According to **Figure 48**, DSR suffers from the highest delay. With OLSR, some packets have higher delays than AODV but the average delay is the lowest with OLSR.

Table 6 shows the minimum, maximum and average delay as well as the median and standard deviation for the three routing protocols.

Table 6. Delay of TCP Packets in Scenario 2.

	Minimum Delay	Maximum Delay	Average Delay	σ	Median
AODV	0.019792458	0.4837698690	0.2206338077	0.073508	0.23909524
DSR	0.010026237	2.1401044990	0.4346695705	0.263983	0.44614249
OLSR	0.010026436	0.4992087260	0.1781333058	0.087942	0.18036488

It can be concluded from **Figure 49** and the standard deviation in **Table 6** that DSR has some packets with extremely high delay (up to 2 seconds) which causes the average delay to be higher. OLSR has the lowest average delay with lowest deviation from that average meaning it is the most robust when it comes to delay and thus safety applications.

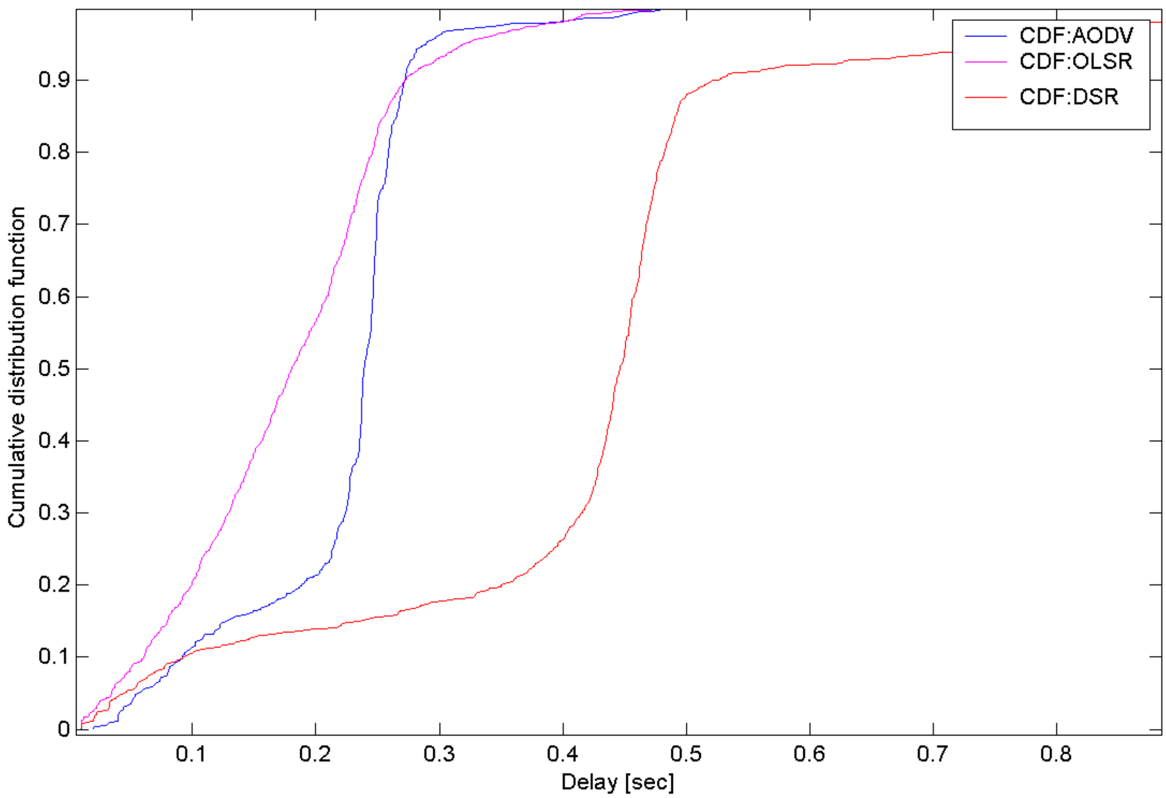


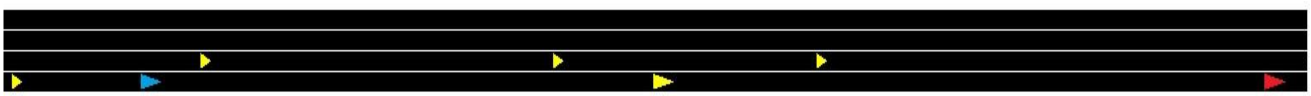
Figure 49. Delay of Scenario 2.

The CDF shows that over than 50% of OLSR packets have less than 0.2 seconds delay versus approximately 12% and 20% for DSR and AODV respectively. With OLSR 80% of sent packets are expected to arrive within 0.24 seconds versus 0.27 seconds for AODV and 0.5 seconds for DSR.

6.3. Scenario 3

The third scenario is designed to test how different routing protocols are affected by interference. The map is a multilane highway with three different flows of cars running at speeds of 20, 22 and 24 m/s and overtaking is allowed. At second 40 the red car starts sending packets to the blue car. At second 70, two other cars (inside red circles) start to communicate causing interference to the other communicating cars (inside by blue circles). At second 100, a third connection is established to add even more interference. The new interfering pair is denoted by green circles. Throughput of received bits and packet delivery ratio (in case of UDP connection) are observed for the car inside the blue circle in the middle of the map. **Figure 50** illustrates car's positions at different time intervals.

Scenario 3 at Second 40



Scenario 3 at Second 70

Scenario 3 at Second 100

Figure 50. Scenario 3 at different time intervals.

Simulation setup:

Transmission time: 90 seconds.

Number of nodes: 21

MAC type: Mac/802_11

Ad-hoc routing protocol: AODV, DSDV, DSR, and OLSR.

Radio Propagation Model: Two Ray Ground.

Transport Protocol: TCP and UDP

Results:

Figure 51 is a comparison of throughput between AODV, DSDV, DSR, and OLSR when using TCP packets.

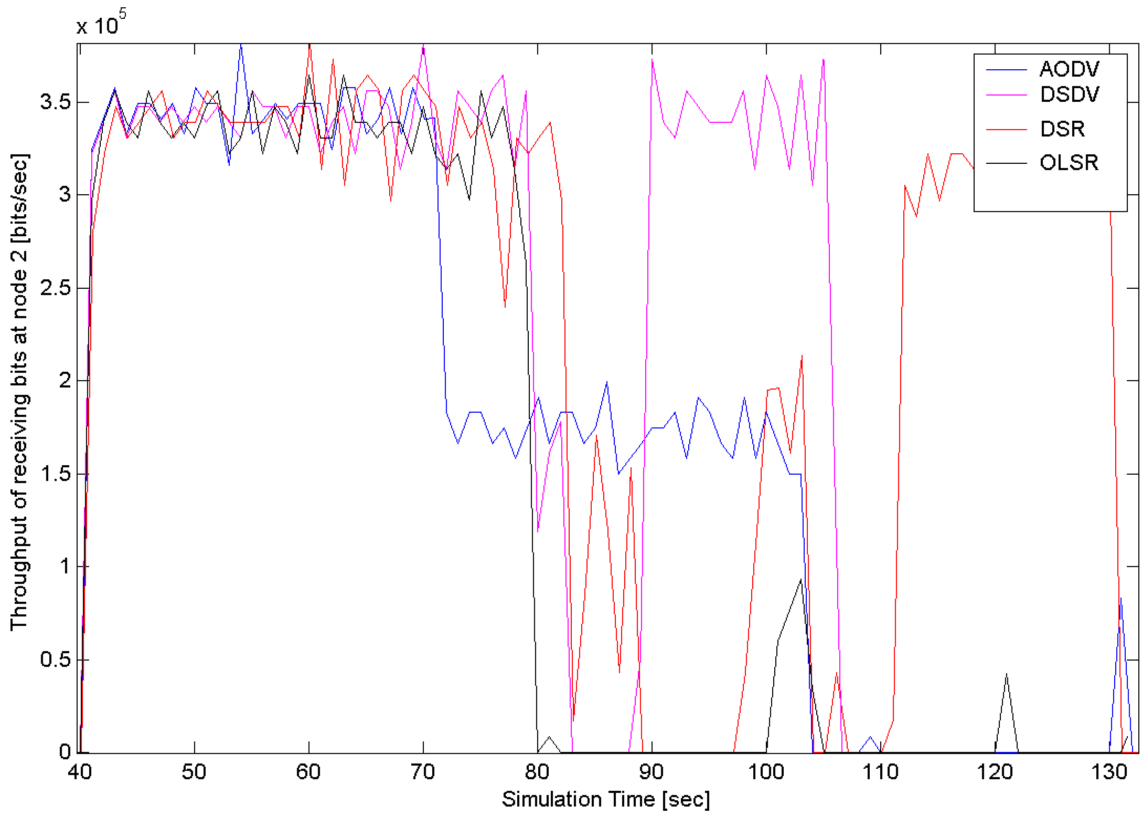


Figure 51. TCP Throughput of Scenario 3.

After second 100, only DSR manages to maintain connection with the presence of two other interfering connections. OLSR shows vulnerability to interference losing connection with only once interference at second 80.

Figure 52 compares throughput with UDP packets.

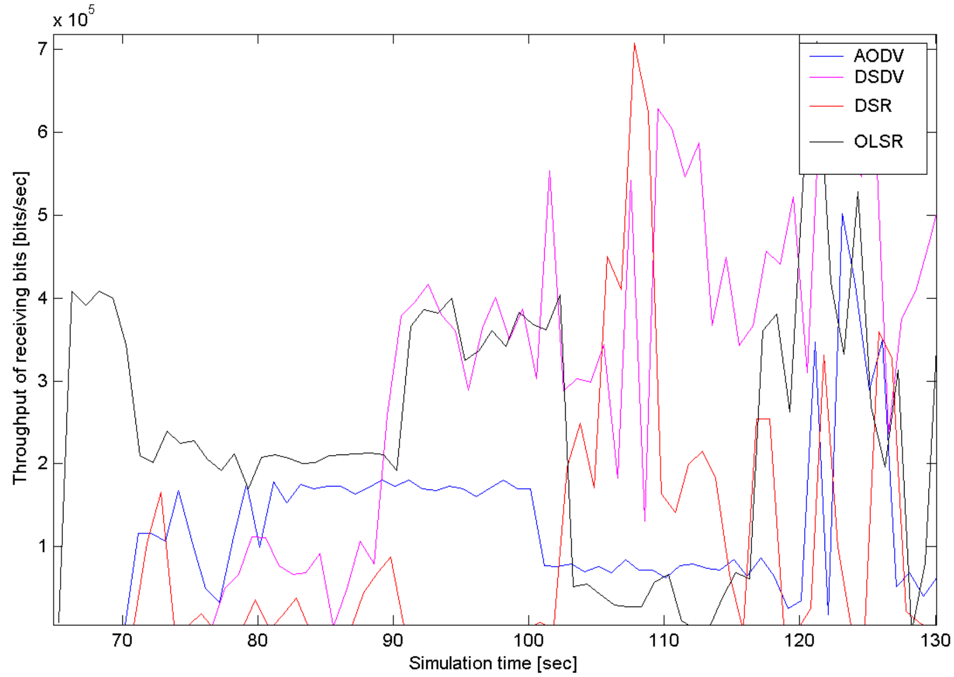


Figure 52. UDP Throughput of Scenario 3.

Results are different with UDP; only OLSR maintains connection throughout the whole simulation time regardless of interference. **Figure 53** compares the average throughput for the four analyzed routing protocols for both UDP and TCP connections as well as the packet delivery ratio for UDP.

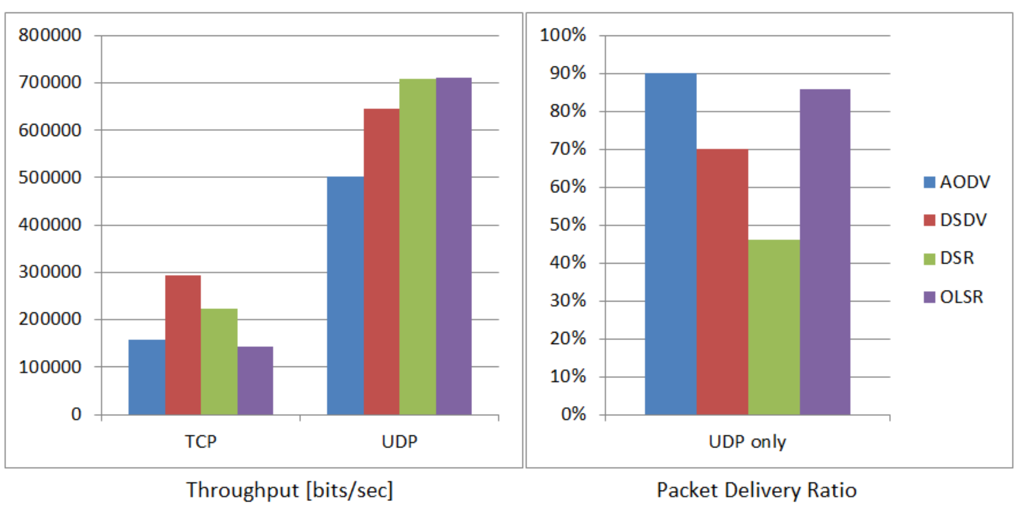


Figure 53. Summary of Scenario 3 Results.

6.4. Scenario 4

This scenario is carried out in a realistic map for the city of Vaasa. A group of cars take a long trip throughout the city starting from a highway in the eastern part and heading towards the church in the city center. The leading car transmits data to its following cars to warn them in case of encountering an obstacle or emergency situation like a stopped car. **Figure 54** shows the path of the trip denoted by a red line.

Figure 54. The Map of Scenario 4

In order to make the scenario more realistic, some of the following cars take turns and leave the main road while new cars join the trip coming from side roads at random points during the trip which induces random changes in the network topology. Throughput, delay and packet delivery ratio are analyzed and compared between different routing protocols at one of the following cars.

Simulation setup:

Transmission time: 190 seconds.

Number of nodes: 15

MAC type: Mac/802_11

Ad-hoc routing protocol: AODV, DSDV, and OLSR.

Radio Propagation Model: Two Ray Ground.

Transport Protocol: TCP and UDP

Results:

Using TCP packets, the three routing protocols achieve approximately the same throughput but DSDV loses connection between second 70 and 120 as shown in **Figure 55**.

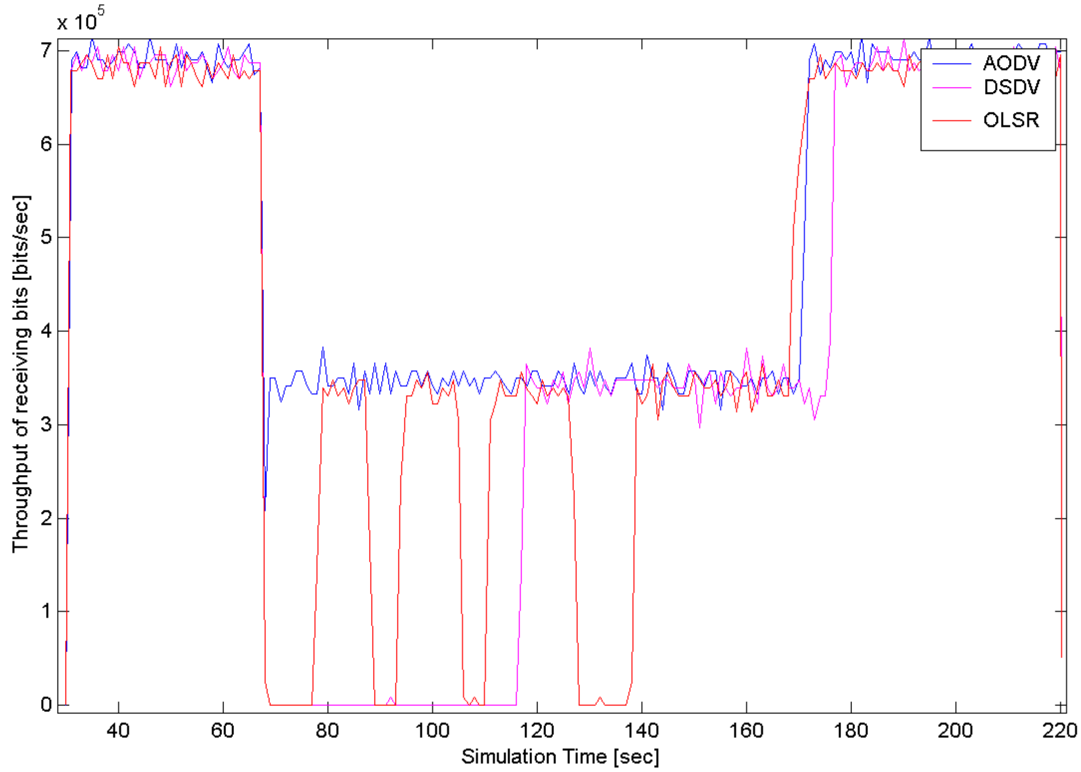


Figure 55. TCP Throughput of Scenario 4.

As for delay, according to **Figure 56**, it does not exceed 0.3 seconds for over 90% of the packets with OLSR having the advantage over DSDV and AODV.

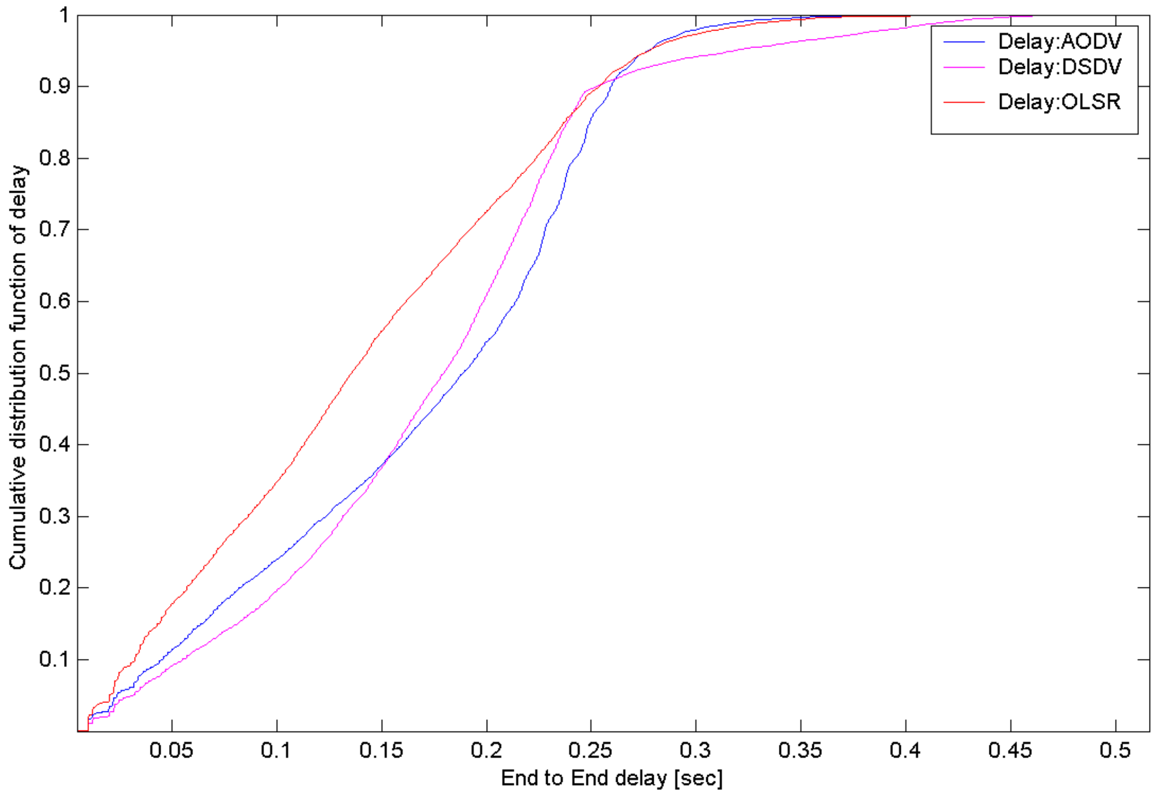


Figure 56. TCP Delay of Scenario 4.

Similarly for UDP packets, throughput is approximately the same but DSDV suffers from longer disconnection periods this time as shown in **Figure 57**. **Figure 58** shows that The three routing protocols have similar CDF of delay; over 60% of packets with delay less than or equal to 0.5 seconds and over 90% of packets with delay less than or equal to 1.1 seconds.

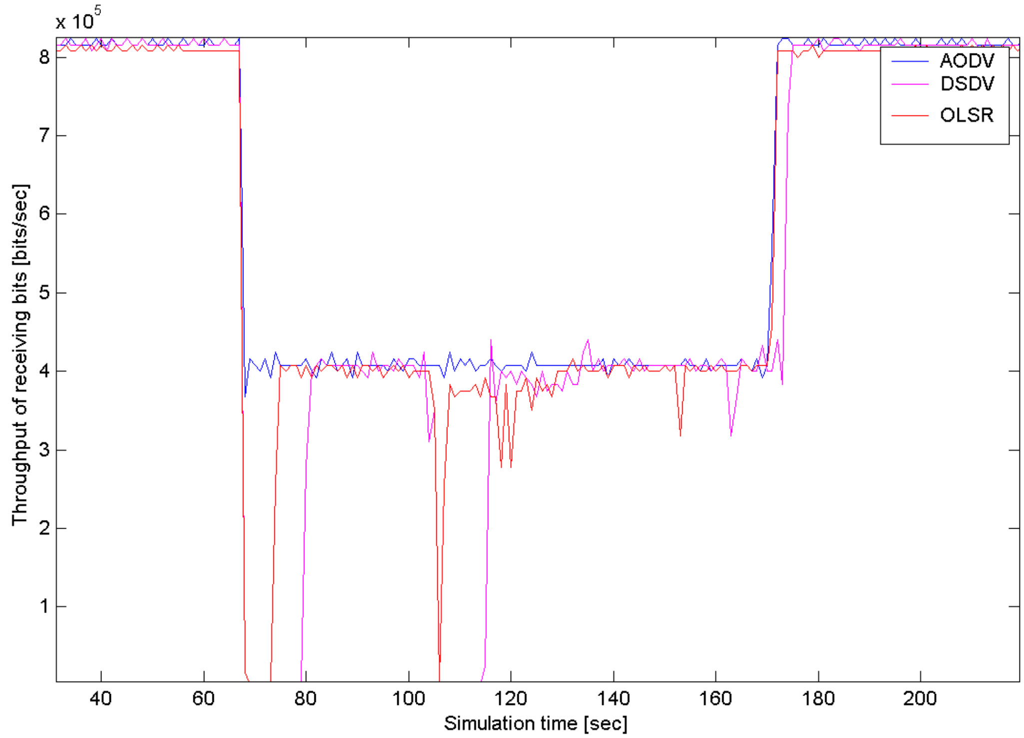


Figure 57. UDP Throughput of Scenario 4

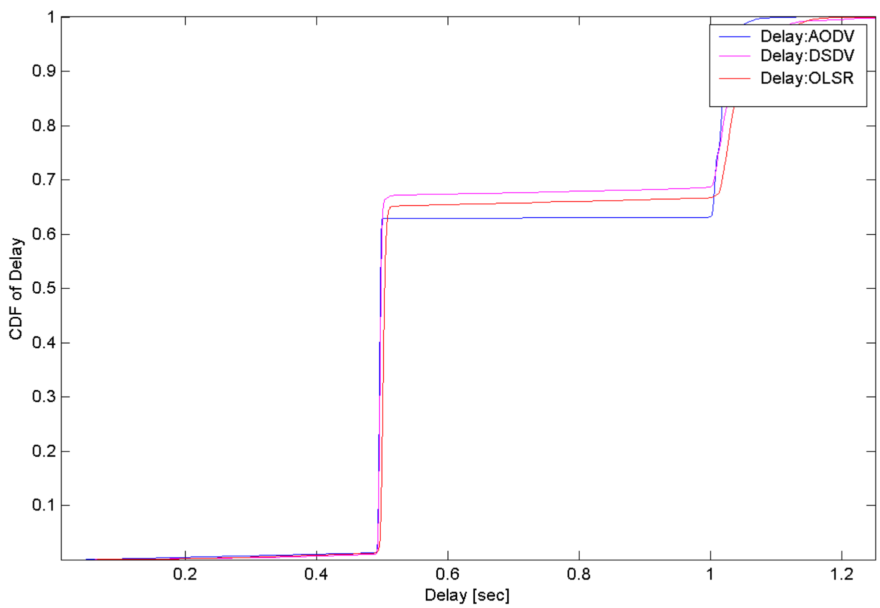


Figure 58. UDP Delay of Scenario 4

Figure 59 compares the average delay, throughput and packet delivery ratio.

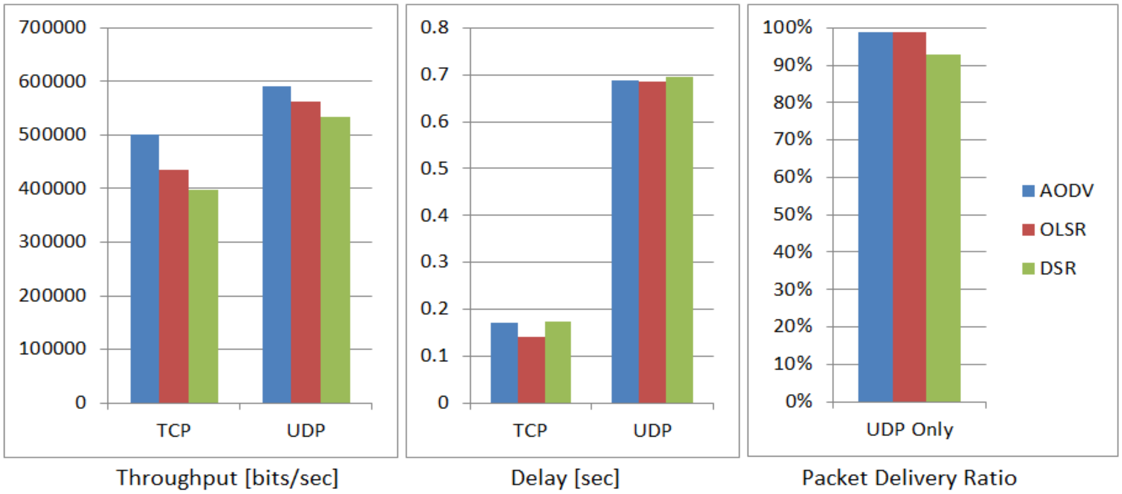


Figure 59. Summary of Scenario 4 Results.

For this scenario, the three tested routing protocols achieve similar results with AODV having slightly higher throughput and OLSR having the lowest delay.

6.5. Scenario 5

This scenario represents an emergency case where a car involved in an accident tries to send a message calling for help to another car (which can be a police car or an ambulance) situated five blocks away (approximately 600 meters). Cars moving throughout the city are used as relays for sending the message. Only TCP is considered. Since the goal is to transmit a full understandable emergency message as fast as possible, the metric for comparison is the total number of successfully received data in bits in a short duration of time (one minute). In **Figure 60**, the car involved in the accident is inside the red circle while the recue car is inside the blue circle.

Figure 60. The Map of Scenario 5.

Simulation setup:

Transmission time: 60 seconds.

Number of nodes: 21

MAC type: Mac/802_11

Ad-hoc routing protocol: AODV, DSR and OLSR.

Radio Propagation Model: Two Ray Ground.

Transport Protocol: TCP.

Results:

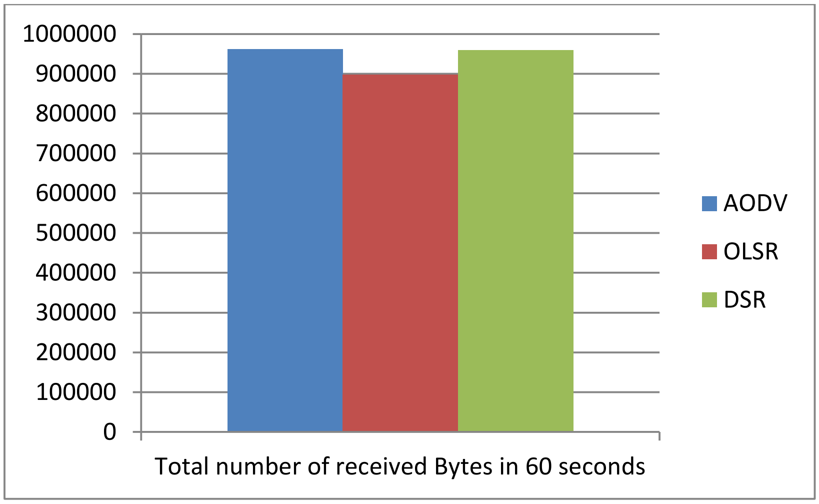


Figure 61. Results of Scenario 5

Due to high mobility in different directions and the large number of hops connecting the source node to the destination, the total amount of received bits is low. It is less than 1MB in one minute according to **Figure 61**. This is enough to send a rescue message defining the location and other data of the car in accident but not enough to establish audio connection.

6.6. Scenario 6a

This scenario compares the effect of different node densities on the performance of the tested routing protocols. A connection is established for 60 seconds between two cars traversing through 4 intersections west of the *Vanha Kirkkopuisto* in Helsinki. Cars density is altered at three steps 57 nodes for low density, 112 nodes for medium density and 167 nodes for high density. **Figure 62** shows the original open street map while **Figure 63** shows the resultant converted SUMO map.

Figure 62. Open Street Map of Scenario 6.

Simulation Setup:

Transmission time: 60 seconds.

Number of nodes: 57, 112 and 167.

MAC type: Mac/802_11

Ad-hoc routing protocol: AODV, DSR and OLSR.

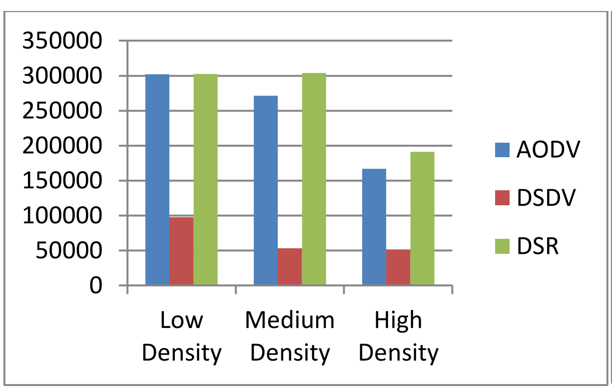
Radio Propagation Model: Two Ray Ground.

Transport Protocol: TCP and UDP.

Figure 63. SUMO Map of Scenario 6.

Results:

Using three routing protocols, two transport protocols and three different densities, a total of 18 simulations are carried out. Results of average delay, throughput and packet delivery ratio are illustrated in **Figure 64**, **Figure 65** and **Figure 66**.



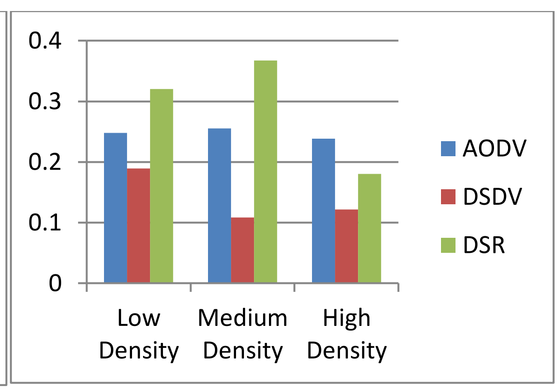


Figure 64. a) Throughput of TCP in Scenario 6.a b) Delay of TCP in Scenario 6.a

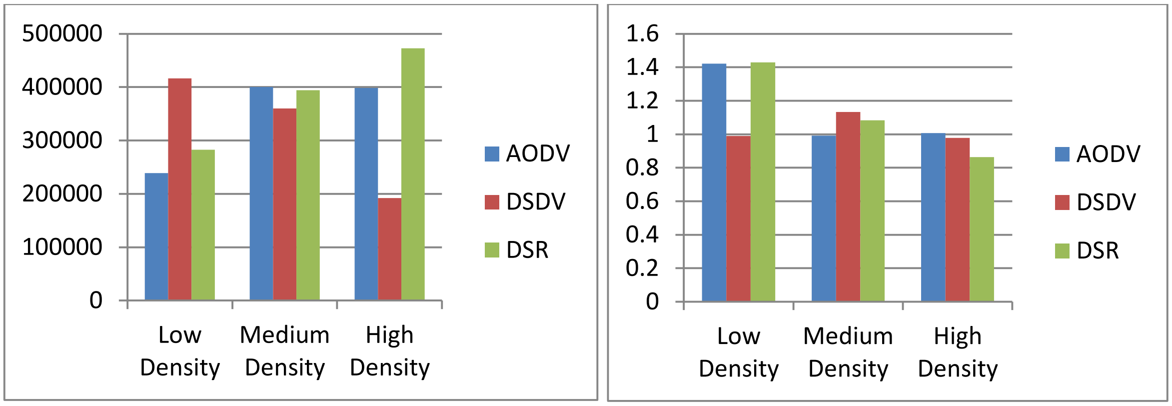


Figure 65. a) Throughput of UDP in Scenario 6.a b) Delay of UDP in Scenario 6.a

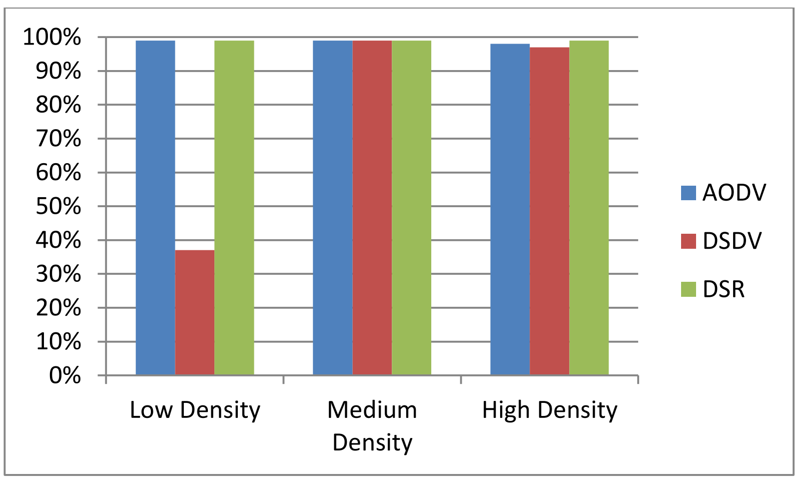


Figure 66. Packet Delivery Ratio of UDP in Scenario 6.a

6.7. Scenario 6b

Same simulations are carried out a second time but with a different pair of nodes and results are shown in **Figures 67**, **Figure 68** and **Figure 69**.

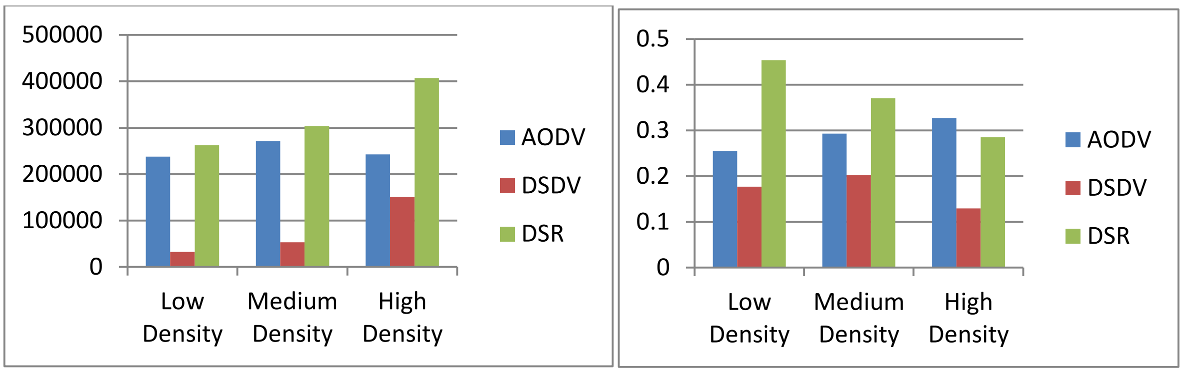


Figure 67. a) Throughput of TCP in Scenario 6.b b) Delay of TCP in Scenario 6.b

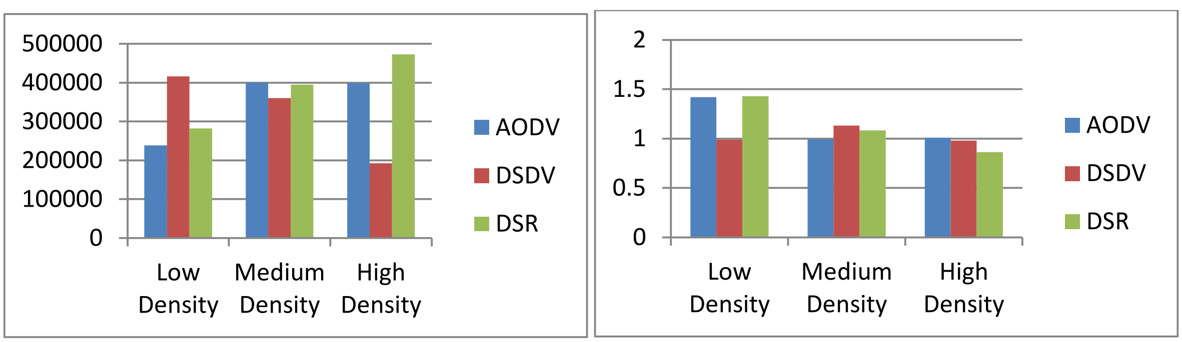


Figure 68. a) Throughput of UDP in Scenario 6.b b) Delay of UDP in Scenario 6.b

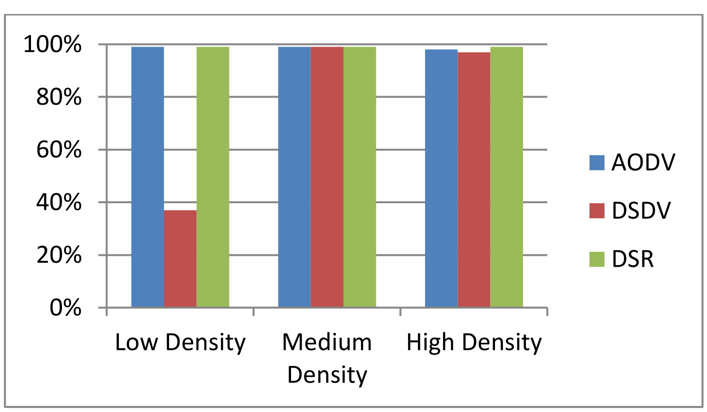


Figure 69. Packet Delivery Ratio of UDP in Scenario 6.b

6.8. Scenario 7

Scenario 7 is a comparison between different wireless channel models. Two nodes follow each other and the leading nodes moves at a faster speed. The leading node transmits to the following node as shown in **Figure 70**. As the distance between the two nodes increases, changes in throughput are observed. The compared wireless channel models are Free Space, Two Ray Ground, and Free Space with Shadowing. Every model is tested with two different transmission powers, first without fading then implementing either Rayleigh or Ricean fading. NCTUns is used for this scenario since those models are already implemented in it.

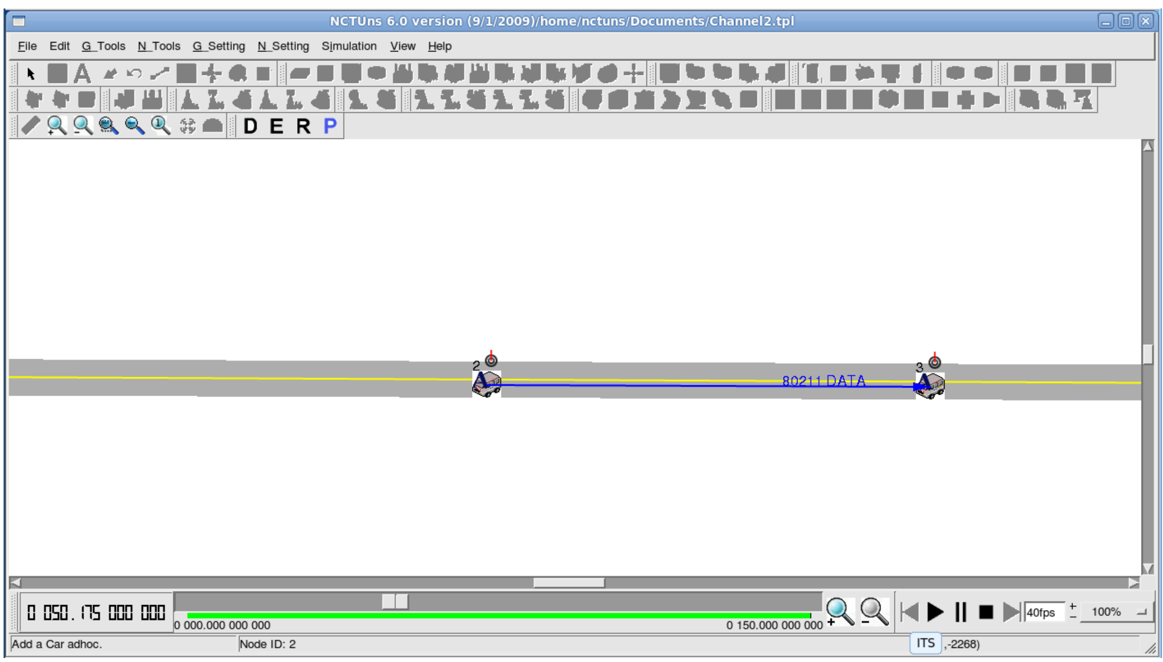


Figure 70. Scenario 7.

Simulation Setup:

Transmission time: 150 seconds.

Number of nodes: 2.

Ad-hoc routing protocol: AODV

Radio Propagation Model: Free Space, Two Ray Ground, Free Space with Shadowing, Rayleigh, and Rician.

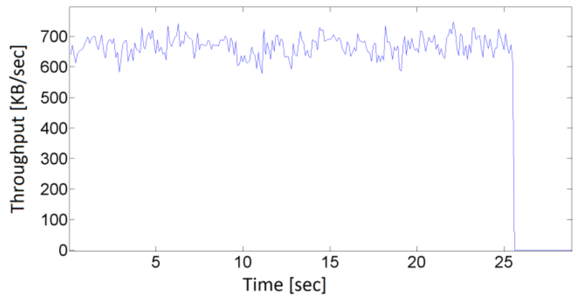
Transport Protocol: TCP.

Transmission Power: 15dBm and 25dBm.

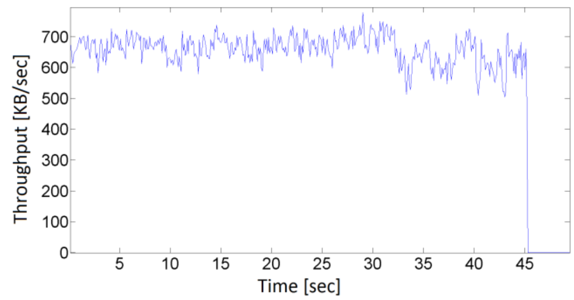
Results:

Figure 71 illustrates throughput results when no fading model is implemented. Throughput is plotted as a function of time. However, it is possible to calculate the distance between the two nodes at any given time from their (x, y) coordinates in the NCTUns GUI.

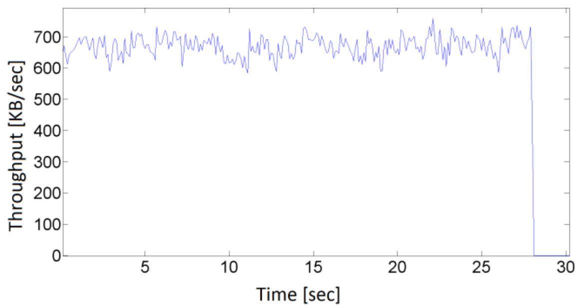
In both free space model and Two Ray Ground model, throughput has a constant average of 650 KB/s and then it goes suddenly to zero once the distance between the two nodes is large enough to cause the received power to fall below the minimum threshold which detectable by the receiver. Transmitting at 15dBm, Free Space Model loses connection at a distance of 250m while Two Ray Ground gets disconnected at a distance of 250m. Increasing the transmission power from 15dBm to 25dBm increases the transmission range greatly up to 860 meter for free space model. Two Ray Ground has a range of 440m in case of 25dBm transmission power, an increase proportional to the increase of transmission power. These results correspond with theoretical assumptions; they show that Two Ray Ground model offers more accurate results at longer distance because it considers reflection via ground. With Free Space and Shadowing model, throughput starts to degrade gradually till it reaches zero at a distance of 315m and 1370m for 15dBm and 25dBm transmission power respectively. More intense fluctuations can be observed in the throughput because the shadowing model considers the multipath effect. Fluctuations get more intense at longer distances because of the absence of the constant part accounting for the LOS received signal.



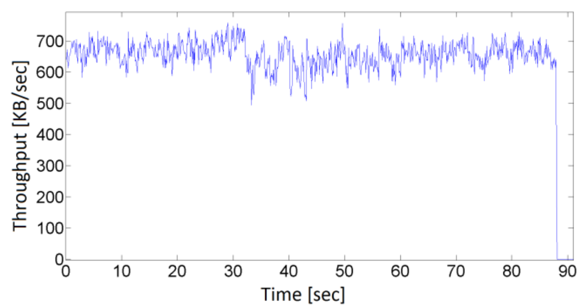
Two Ray Ground, 15dBm



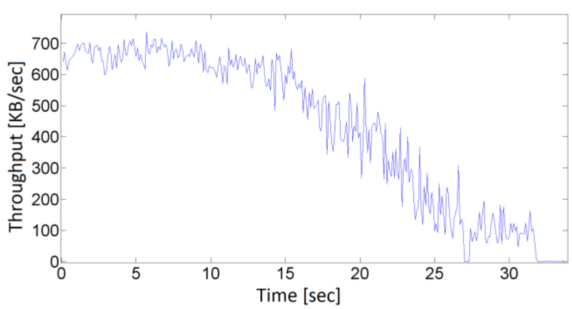
Two Ray Ground, 25dBm



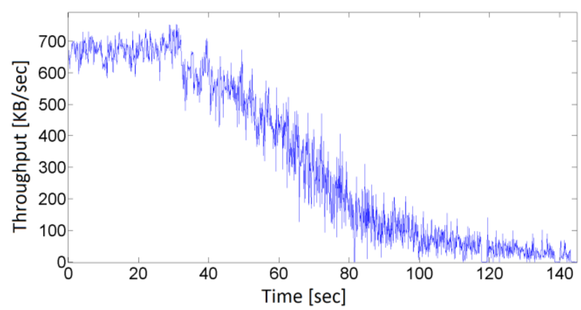
Free Space 15dBm



Free Space 25dBm



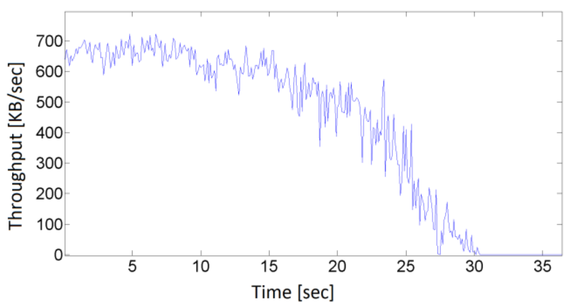
Free Space and Shadowing 15dBm



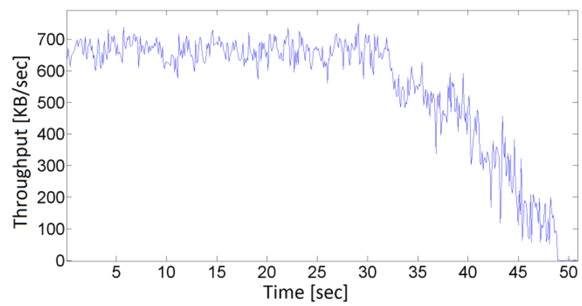
Free Space and Shadowing 25dBm

Figure 71. Throughput of Scenario 7 with no fading

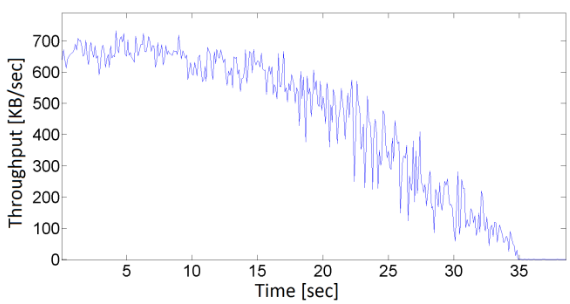
Rayleigh fading assumes the absence of a line of sight component and considers only the multipath variations. Adding Rayleigh fading to either Free Space or Two Ray ground models produced results similar to Free Space and Shadowing model as shown in **Figure 72.** Adding Rayleigh fading to the Free Space and Shadowing model produced extremely intense variations in the received throughput.



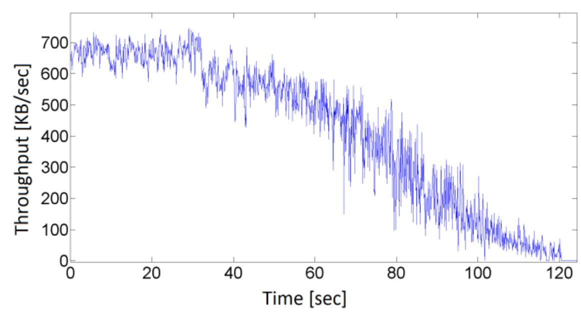
Two Ray Ground, 15dBm



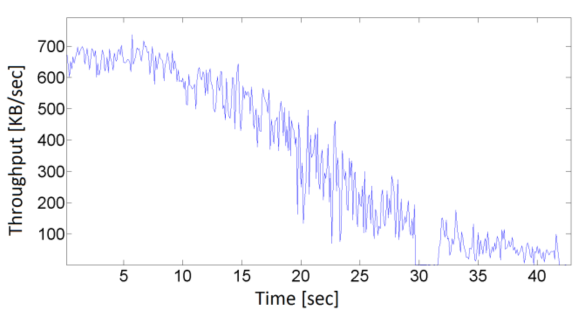
Two Ray Ground, 25dBm



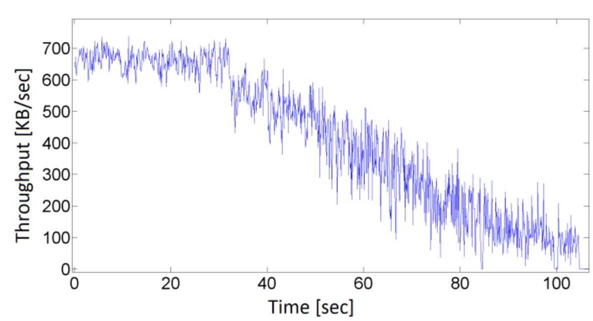
Free Space 15dBm



Free Space 25dBm



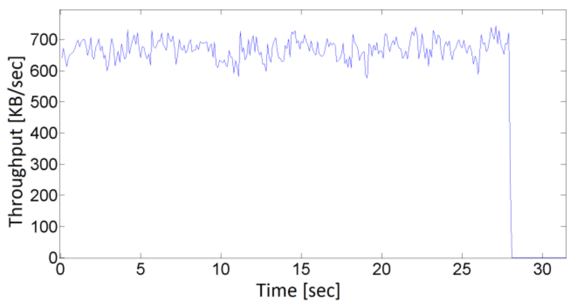
Free Space and Shadowing 15dBm



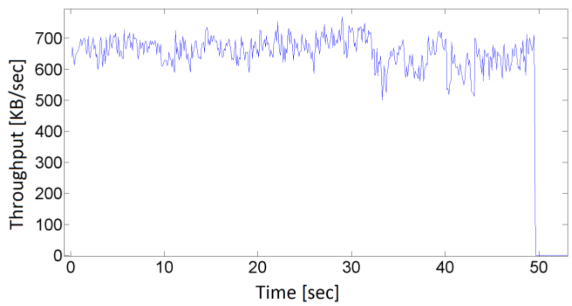
Free Space and Shadowing 25dBm

Figure 72. Throughput of Scenario 7 with Rayleigh Fading

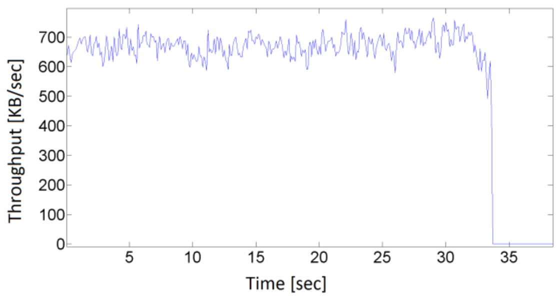
A Rician factor of k=10 was used in this experiment which emphasized the LOS component. A slight increase in the transmission range can be observed in **Figure 73** for the three tested wireless channel models.



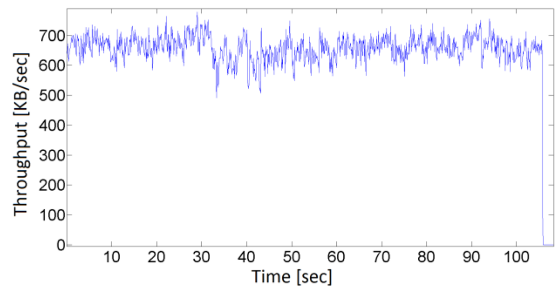
Two Ray Ground, 15dBm



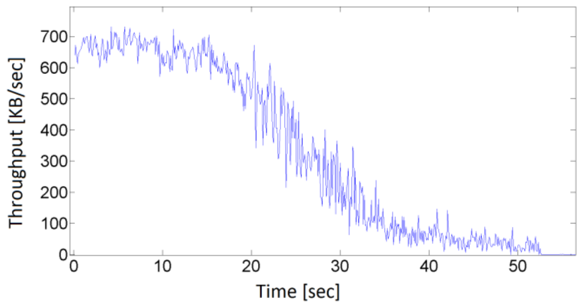
Two Ray Ground, 25dBm



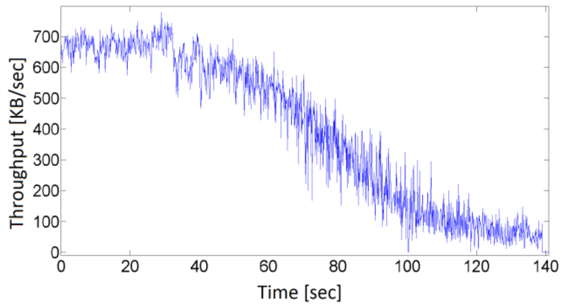
Free Space 15dBm



Free Space 25dBm



Free Space and Shadowing 15dBm



Free Space and Shadowing 25dBm

Figure 73. Throughput of Scenario 7 with Rician Fading

7. CONCLUSION AND FUTURE WORK

Based on reviewed scientific papers as well as the simulations done in this thesis, it is concluded that until today, there is no dedicated VANET simulator capable of satisfying researcher's needs. All of the surveyed VANET simulators were originally designed to simulate MANETs and were modified later to incorporate some VANET functionalities. Another approach is to use a middleware to couple a powerful mobility simulator with a common network simulator such as the combination of ns-2 and SUMO coupled by MOVE which was used in this thesis simulations. This approach however, fails to consider the effect of received messages and warnings on the driver's behavior and further complicates the simulation process. Therefore, it is strongly recommended to work on creating dedicated VANET simulators. Some of the desired characteristics of such simulators are the ability to load maps including their traffic regulations from the internet directly, simulate a wider array of VANET routing protocols and produce random realistic mobility of cars based on the data obtained from real traffic surveys.

Part of the performed simulations in chapter 6 used realistic city maps and completely random node movement to simulate real world scenarios. The other simulations were synthesized and tailored to test the impact of specific parameters on the performance of VANET networks.

Comparisons between four different routing protocols (AODV, DSR, DSDV and OLSR) were carried out using TCP and UDP packets and the evaluation metrics used were the throughput, end to end delay and packet delivery ratio.

According to the results of those simulations, AODV is the most robust routing protocol with highest throughput and packet delivery rate and lowest disconnection time in the majority of the performed simulations. DSDV provided lowest delays but suffers from lower throughput and more occasional disconnections. OLSR achieved similar results to

AODV with slightly lower delays and throughput while AODV maintains the advantage in connection stability. DSR provides high throughput but suffers from the highest delay and least consistent connection.

Simulation results showed also differences between the performance of TCP and UDP transport protocols. While UDP achieves higher throughput, it suffers from much higher delays. This problem along with the packet drop makes UDP unsuitable for safety applications. In future work, more detailed comparisons between TCP and UDP in VANETs will be performed.

Scenario 7 tested several wireless channel models available in the NCTUns simulator. The obtained results correspond to theoretical expectations. However, it is not possible to conclude which channel model gives the best approximation for real world VANET networks without the availability of field measurements. A further future work could be to perform field experiments or to collect data from other researcher's experiments and replicate those experiments in VANET simulators. The data obtained from field measurements will be compared with results obtained from simulators in order to conclude which wireless channel model can produce the best approximation.

REFERENCES

Allal, Salim & Saadi Boudjit (2012). *Geocast Routing Protocols for VANETs: Survey and Guidelines*. In: 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 4–6 July 2012, Palermo. IEEE Print ISBN 978-1-4673-1328-5. Available from Internet: <URL:http://isyou.info/jisis/vol3/no12/jisis-2013-vol3-no12-02.pdf >.

Bustamante, Fabian E (2013). *STRAW - STreet RAndom Waypoint - vehicular mobility model for network simulations* [online]. Available from Internet: <URL: http://www.aqualab.cs.northwestern.edu/projects/144-straw-street-random-waypoint-vehicular-mobility-model-for-network-simulations-e-g-car-networks>.

Chou , Chine-Ming (2011). *Rapid Generation of Realistic Simulation for VANET Manual.* Available from Internet:<URL: http://lens1.csie.ncku.edu.tw/MOVE/Example%20for%20step-by-step.pdf>.

Eenennaam , E.M. van (2009). *A Survey of Propagation Models used in Vehicular Ad hoc Network (VANET) Research*. Netherland, University of Twente, Faculty of EEMCS, Design and Analysis of Communication Systems group. Available from Internet: <URL: http://wwwhome.ewi.utwente.nl/~eenennaa/unpublished/vanEenennaam_MRC_paper.pdf>.

Elmusrati, Mohammed (2011). *Radio Resource Management (TLTE3050) Course Slides*. University of Vaasa, Finland.

Fiore, Marco, Jerome Harri, Fethi Filali & Christian Bonnet (2007).*Vehicular Mobility Simulation for VANETs.* In: 40th Annual Simulation Symposium ANSS '07, 2007, Norfolk, VA. ISSN: 1080-241X. ISBN 978-1-4244-1513-7.

Gangurde Pallavi, Sanjay Waware & Nisha Sarwade (2013). *Simulation of TCP, UDP and SCTP with constant traffic for VOIP services.* International Journal of Engineering Research and Applications (IJERA), 2:3, 1245–1248. ISSN: 2248-9622.

Giannoulis , Spilios, Christos Antonopoulos, Evangelos Topalis, Antonis Athanasopoulos Aggeliki Prayati & Stavros Koubias (2006). *TCP vs UDP performance evaluation for CBR traffic on Wireless Multihop Networks*. In: Proceedings of CSNDSP 2006, 5th International Symposium on Communication Systems, Networks and Digital Signal Processing, 154–158. July 2006, Patras, Greece. ISBN: 960-89282-0-6. Available from Internet:<URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.5517&rep=rep1&type=pdf>.

Goyal, Priyanka, Vinti Parmar & Rahul Rishi (2011). *MANET: Vulnerabilities, Challenges, Attacks, Application.* IJCEM International Journal of Computational Engineering & Management [online]. Available from Internet: <URL: http://skirubame.ucoz.com/_ld/0/29_Topic_1-MANET_v.pdf>. ISSN 2230-7893.

Härri, Jerome, Fethi Filali & Christian Bonnet (2009). *Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy.* IEEE Communications Surveys & Tutorials, 11:4, 19–41. ISSN:1553-877X.

Hartenstein, Hannes & Kenneth P. Laberteaux (2010). *VANET: Vehicular Applications and Inter-Networking Technologies.* 1st Ed. Wiley. 435 p. ISBN 978-0-470-74056-9.

Hui, Kwan Lim & Amitava Datta (2012). *Enhancing the TORA Protocol using Network Localization and Selective Node Participation*. In: 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC), 9-12 Sept. 2012, Sydney. ISSN: 2166-9570.

Jangra, Ajay, Nitin Goel, Priyanka & Komal Kumar Bhahtia (2010). *IEEE WLANs Standards for Mobile Ad-hoc Networks (MANETs): Performance Analysis*. Global Journal of Computer Science and Technology, 10:14, 42–47. Available from Internet: <URL: http://www.academia.edu/541696/IEEE_WLANs_Standards_for_Mobile_Ad-hoc_Networks_MANETs_Performance_Analysis>.

Karp, Brad & H. T. Kung (2000). *GPSR: Greedy Perimeter Stateless Routing for Wireless Networks.* In: MobiCom '00 Proceedings of the 6th annual international conference on Mobile computing and networking. New York, USA. ACM ISBN:1-58113-197-6. Available from Internet:<URL: http://www.eecs.harvard.edu/~htk/publication/2000-mobi-karp-kung.pdf>.

Kenney, John B (2011). *Dedicated Short-Range Communications (DSRC) Standards in the United States.* Proceedings of the IEEE, 99:7, 1162-1182. ISSN: 0018-9219.

Khairnar, Vaishali D. & S.N. Pradhan (2011). *Mobility models for Vehicular Ad-hoc Network simulation*. In: 2011 IEEE Symposium on Computers & Informatics (ISCI). Kuala Lumpur: IEEE 978-1-61284-689-7. Available from Internet: <URL: http://arxiv.org/ftp/arxiv/papers/1304/1304.5189.pdf>.

Koivo, Heikki & Mohammed Elmusrati (2009). *Systems Engineering in Wireless Communications.* 1st Ed. Wiley. 341 p. ISBN 978-0-470-02178-1.

Labiod, Houda (2008). *Wireless Ad Hoc and Sensor Networks*. 2nd Ed. London: ISTE Ltd. 317 p. ISBN 978-1-84821-003-5.

Laboratory for Experimental Network and System (LENS). *Rapid Generation of Realistic Simulation for VANET (2012)*. Available from Internet: <URL: http://lens.csie.ncku.edu.tw/Joomla_version/index.php/research-projects/past/18-rapid-vanet>.

Lee, Kevin, Uichin Lee & Mario Gerla (2009). *Survey of Routing Protocols in Vehicular Ad Hoc Networks. Advances in Vehicular Ad-Hoc Networks: Developments and Challenges*. IGI Global, (Oct, 2009). Available from Internet: <URL: http://netlab.cs.ucla.edu/wiki/files/RoutingBookChapterKLULMario.pdf>.

Lin, Yun-Wei , Yuh-Shyan Chen & Sing-Ling Lee (2010). *Routing Protocols in Vehicular Ad Hoc Networks: A Survey and Future Perspectives*. Journal of Information Science and Engineering 26, 913–932 (2010). Available from Internet: <URL: http://www.psut.jo/sites/raad/wireless_notes/VANET.pdf>.

Liu, K. J. Ray, Ahmed K. Sadek, Weifeng Su & Andres Kwasinski (2008). *Cooperative Communications and Networking. 1st Ed. Cambridge University Press.* 627 p. ISBN-13: 978-0-521-89513-2.

Lopez , Jesus Gabriel Balderas (2010). *A survey of geographic routing protocols for Vehicular Ad Hoc Networks (VANETs)* [online]. New Mexico State University, Fall 2010 CS 479/579. Available from Internet: <URL: http://www.slideshare.net/gabitobalderas/a-survey-of-geographic-routing-protocols-for-vehicular-ad-hoc-networks-vanets>.

Mahajan, Atulya, Niranjan Potnis, Kartik Gopalan & An-I A. Wang (2006). *Urban Mobility Models for VANETs*. In: proceedings of second workshop on next generation wireless networks. Available from Internet: <URL: http://www.cs.binghamton.edu/~kartik/publications/mahajan06urban.pdf>.

Martinez, Francisco, Chai Keong Toh, Juan-Carlos Cano, Carlos T. Calafate & Pietro Manzoni (2009). *A survey and comparative study of simulators for vehicular ad hoc networks (VANETs).* Wireless Communications and Mobile Computing. Published online in Wiley InterScience, Volume 11, Issue 7, pages 813–828, July 2011.

Martinez, Francisco J., Juan-Carlos Cano, Carlos T. Calafate & Pietro Manzoni. *CityMob: a mobility model pattern generator for VANETs*. IEEE International Conference on Communications Workshops, 2008. ICC Workshops '08, Beijing. Pages 370 – 374. IEEE Print ISBN 978-1-4244-2052-0.

Misra , Padmini (2000). *Routing Protocols for Ad Hoc Mobil Wireless Networks* [online]. Available from Internet: <URL: http://www.cse.wustl.edu/~jain/cis788-99/ftp/adhoc_routing.pdf>.

Nagaraj, Uma & Poonam Dhamal (2011a). *Broadcasting Routing Protocols in VANET. International Institute for Science, Technology & Education*, 1:2, 13–19. ISSN 2224-610X. Available from Internet: <URL: http://www.iiste.org/Journals/index.php/NCS/article/view/1245/1166>.

Nagaraj, Uma, M. U. Kharat & Poonam Dhamal (2011b). *Study of Various Routing Protocols in VANET*. International Journal of Computer Science and Telecommunications, 2:4, 45–52. Available from Internet: <URL: http://www.ijcst.com/vol24/1/umanagaraj.pdf>. ISSN : 0976-8491.

Narra, Hemanth, Yufei Cheng, Egemen K. Cetinkaya, Justin P. Rohrer. & James P.G. Sterbenz (2011). *Destination-sequenced distance vector (DSDV) routing protocol implementation in ns-3*. In: Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, Brussels, Belgium. ISBN: 978-1-936968-00-8.

National Instruments (2013). *Pulse-Shape Filtering in Communications Systems*. Available from Internet:<URL: http://www.ni.com/white-paper/3876/en/pdf>.

Naumov, Valery, Rainer Baumann & Thomas Gross (2006). *An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces*. In: MobiHoc'06, May 22–25, 2006, Florence, Italy. Copyright 2006 ACM 1-59593-368-9/06/0005. Available from Internet: <URL: http://www2.ic.uff.br/~celio/classes/cmovel/slides/vanet-acm06.pdf>.

Paul, Bijan, Md. Ibrahim & Md. Abu Naser Bikas (2011). *VANET Routing Protocols: Pros and Cons.* International Journal of Computer Applications 20:3, 28–34. Available from Internet: <URL: http://arxiv.org/ftp/arxiv/papers/1204/1204.1201.pdf.

Piorkowski, Michał, Maxim Raya, Ada Lezama Lugo, Panos Papadimitratos, Matthias Grossglauser & JeanPierre Hubaux (2008). *TraNS: Realistic Joint Traffic and Network Simulator for VANETs*. ACM SIGMOBILE Mobile Computing and Communications Review, 12:1, 31-33. Available from Internet: <URL: http://infoscience.epfl.ch/record/113879/files/trans_mc2r_2007.pdf>.

Pipe, L (1953). *An Operational Analysis of Traffic Dynamics*. Journal of Applied Physics 24, 274–281.

Rappaport, Theodore (2002). *Wireless Communications: Principles and Practice*. 2nd Ed. Prentice Hall. 707 P. ISBN 978-0-130-42232-3.

Rastogi, Devashish, Sachin Ganu, Yanyong Zhang, Wade Trappe & Charles Graff (2007). *A Comparative Study of AODV and OLSR on the ORBIT Testbed*. IEEE Military Communications Conference, MILCOM, Orlando FL, USA, Oct. 2007. ISBN 978-1-4244-1513-7.

Raw, Shringar Ram & Sanjoy Das (2011). *Performance Comparison of Position-Based Routing Protocols in Vehicle-to-Vehicle (V2V) Communication*. International Journal of Engineering Science and Technology, 3:1, 435–444. Available from Internet:<URL: http://www.ijest.info/docs/IJEST11-03-01-118.pdf>. ISSN : 0975-5462.

Rhattoy, A. & A. Zatni (2012).*The Impact of Radio Propagation Models on Ad Hoc Networks Performances.* Journal of Computer Science, 8:5,752–760. ISSN 1549-3636.

Santos, R. A., O. Álvarez & A. Edwards (2005). *Performance evaluation of two location-based routing protocols in vehicular Ad-Hoc networks.* IEEE 62nd Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005. ISSN: 1090-3038.

Singh, Pranav Kumar & Kapang Lego (2011). *Comparative Study of Radio Propagation and Mobility Models in Vehicular Adhoc Network.* International Journal of Computer Applications, 16:8, 37-42. Available from Internet: <URL: http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.206.2898>.

Sommer, Christoph (2012). Veins documentation [online]. Available from Internet: <URL: http://veins.car2x.org/documentation/>.

Spaho, Evjola, Leonard Barolli, Gjergji Mino, Fatos Xhafa & Vladi Kolici (2011). *VANET Simulators: A Survey on Mobility and Routing Protocols.* Broadband and Wireless Computing, Communication and Applications (BWCCA). IEEE 978-1-4577-1455-9.

SUMO (2013). *Sumo at a Glance* [online]. Available from Internet: <URL: http://sumo-sim.org/wiki/Sumo_at_a_Glance>.

Taleb, Tarik, Ehssan Sakhaee, Abbas Jamalipour, Kazuo Hashimoto, Nei Kato & Yoshiaki Nemoto (2007). *A Stable Routing Protocol to Support ITS Services in VANET Networks*. IEEE Transactions on Vehicular Technology, 56:6, 3337- 3347. ISSN: 0018-9545.

Varga, András (2013). *What is OMNeT++?* [online]. Available from Internet: <URL: http://www.omnetpp.org/home/what-is-omnet>.

Wang, Shie-Yuan & Chih-Liang Chou .*NCTUns Simulator for Wireless Vehicular Ad Hoc Network Research.* Available from Internet: <URL: http://nsl.csie.nctu.edu.tw/files/NovaVANET.pdf>.

Xia, Weiwei, Tiecheng Song & Lianfeng Shen (2010). *A Cluster-Based Directional Routing Protocol in VANET*. IEEE International Conference on Communication Technology (ICCT), 2010 12th, Nanjing, China. Print ISBN 978-1-4244-6868-3.

Yousefi , Saleh, Mahmoud Siadat Mousavi & Mahmood Fathy (2006). *Vehicular Ad Hoc Networks (VANETs) Challenges and perspectives*. International Journal of Computer Applications (0975 – 8887) 20:3, 761–766.