Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FINAL DE CARRERA

PERFORMANCE EVALUATION OF REALISTIC SCENARIOS FOR VEHICULAR AD HOC NETWORKS WITH VANETMOBISIM AND NS2 SIMULATORS

Estudis: Enginyeria de Telecomunicació

Autor: Roger Calzada Cañero

Directora: Mónica Aguilar Igartua

Co-director: Ahmad Mohamad Mezher

Any: Juny 2011

AGRAÏMENTS

En primer lloc m'agradaria donar les gràcies a la Mónica, pel seu enorme interés i orientació durant la realització d’aquest projecte.

Es mereixen una especial menció el Diego, el Xavi, la Carolina i l'Ahmad, per la seva essencial col·laboració i aportació al llarg d'aquests darrers mesos. A tots quatre, us desitjo la millor de les sorts en el futur.

Finalment, a la meva família per l'inestimable suport rebut al llarg de la meva etapa universitària que ara culmina.

I sobretot a la Bea, pel seu recolzament i comprensió en tot moment.

INDEX

INDEX OF FIGURES AND TABLES

Figures

Tables

GLOSSARY OF ACRONYMS

ACK	ACKnowledgment
ADAS	Advanced Driver Assistance Services
ADV	Adaptative Distance Vector
AIFS	Arbitration Interframe Space
AODV	Ad-hoc On-demand Distance Vector
AWK	Aho Weinberg Brian
BSS	Basis Service Set
BPSK	Binary Phase-Shift Keying
CALM	Communication Architecture for Land Mobile environment
CANUMOBISIM	Communications in Ad Hoc Networks for Ubiquitous computing Mobility Simulator
CBR	Constant Bit Rate
CCH	Control Channel
CW	Contention Window
DARPA	Defense Advanced Research Projects Agency
DCF	Distributed Coordination Function
DNT	Delay Tolerant Network
DSDV	Dynamic Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
DSSS	Direct-Sequence Spread Spectrum
DSRC	Dedicated Short-Range Communications

E2E	End-to-End
ECC	Electronic Communications Committee
EDCA	Enhanced Distributed Channel Access
EIRP	Equivalent Isotropically Radiated Power
FCC	Federal Communication Commission
GDF	Geographic Data Files
GSM	Global System for Mobile communications
GTS	Global System of Telematics
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronic Engineers
IDM	Intelligent Driving Model
IDM_IM	Intelligent Driving Model with Intersection Management
IDM_LC	Intelligent Driving Model with Lane Changing
IMM	Integrated Mobility Model
IP	Internet Protocol
ISM	Industrial Scientific Medial
ISO	International Organization for Standardization
ITS	Intelligent transportation system
LAGAD	Location-Aided Gateway Advertisement and Discovery
LAN	Local Area Network
LOS	Line of Sight
MAC	Medium Access Control
MANET	Mobile Ad hoc NETwork
MLME	MAC Layer Management Entity
MOVE	MObility model generator for VEhicular networks
NCTUns	National Chiao Tung University Network Simulator
NS2	Network Simulator
OBU	On Board Unit
OFDM	Orthogonal Frequency-Division multiplexing
OLSR	Optimized Link State Routing
OTCL	Object-oriented Tool Command Language
PAN	Personal Area Network
PDA	Personal Digital Assistant
PDF	Probability Density Function

PLME	Physical Layer Management Entity
POI	Point Of Interest
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase-Shift Keying
RDPR	Received Data Packet Ratio
RERR	Router Error
RREP	Route Reply
RREQ	Route Request
RSE	Road Side Equipment
RSU	RoadSide Unit
SCH	Service Channel
SUMO	Simulation of Urban Mobility
TCL	Tool Command Language
TCP	Transfer Control Protocol
TIGER	Topologically Integrated Geographic Encoding and Referencing
TTL	Time To Live
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
V2V	Vehicle to Vehicle
VANET	Vehicular Ad-hoc NETwork
VANETMOBISIM	Vehicular Ad-hoc NETwork Mobility Simulator
VINT	Virtual InterNetwork Testbed
WAVE	Wireless Access for Vehicular Environment
WBSS	WAVE Basis Service Set
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
XML	eXtensible Markup Language
ZRP	Zone Routing Protocol

ABSTRACT

Over recent years, the considerable mobile services sector growth around the world was certainly the major phenomenon in the telecommunications field. Wireless technology has led to the development of new communications systems and multimedia services. Due to the continued growth of the vehicular industry and the increasing demand of road safety, a new concept in the communications field was born: vehicular networks (VANETs). In VANETs, each vehicle could act as router or node, establishing connections among nearby vehicles or with roadside infrastructure.

VANETs are receiving more attention from governments and car manufacturers due to the wide variety of applications and services they can provide such as road safety systems, car assistance and Internet acces. However, designing and implementing VANETs is a complex and wide area of research as we can notice, knowing that in the last years the research and development community has focused on the study of such networks.

Basically, our project is divided in two main parts: Firstly, we made a state of art related to the actual state of VANETs nowadays in order to find the most appropiate and recommended mobility generator and network simulator reported in the literature. Secondly, we decided to use VanetMobiSim [80], as a mobility generator due to its variety mobility models that could be tested, and NS2 [63] as a network simulator for being one of the most used by many authors and also due to its compatibilty with VanetMobiSim. Using these tools, VanetMobiSim and NS2, we carried out a deep performance evaluation of VANETs in several realistic scenarios, giving different values to parameters such as the number of nodes, speed and the propagation model.

RESUM

En els darrers anys, el considerable creixement del sector dels serveis mòbils arreu del món es certament el major fenòmen al camp de les telecomunicaciones. Les tecnologies inalàmbriques han conduït al desenvolupament de nous sistemes de comunicació y serveis multimèdia. Degut al constant creixement del mercat automobilístic juntament amb la creixent demanda de la seguretat viària ha nascut un nou concepte al camp de les comunicaciones: les xarxes entre vehicles (VANETs). A les VANETs, cada vehicle pot actuar com a router o node, establint connexions entre vehicles propers o amb infraestructura a la carretera.

Les VANET estan rebent més atenció del govern i de la indústria automobilística degut a l'àmplia varietat d'aplicacions y serveis que poden oferir, tal com sistemes de seguretet viària assistència a la carretera i accès a Internet. No obstant, el disseny i l'implementació de VANETs és una àrea d'investigació àmplia i complexa tal i com podem percebre, sabent que durant els darrers anys la comunitat investigadora s'ha centrat en l'estudi d'aquestes xarxes.

Bàsicament, el nostre projecte està dividit en dues parts principals: Primerament, hem dut a terme una recerca relacionada amb l'estat actual de les VANET avui en dia, amb l'objectiu d'identificar els generadors de moviment i els simuladors de xarxes més apropiats i recomenats a la literatura. En segon lloc, hem decidit utilitzar el VanetMobiSim [80], com a generador de moviment degut a la seva varietat de models de movilitat que es poden testejar, i el NS2 [63] com a simulador de xarxes per ser un dels més utilitzats per molts autors a més de la seva compatibilitat amb el VanetMobiSim. Amb l'ús d'aquestes eines, VanetMobiSim i NS2, hem dut a terme una avaluació profunda de les prestacions de les VANET en diversos escenaris reals, assignant valors diferents a paràmetres tals com el nombre de nodes, la velocitat i el model de propagació.

RESUMEN

En los últimos años, el considerable crecimiento del sector de los servicios móviles alrededor del mundo es con certeza el mayor fenómeno en el campo de las telecomunicaciones. Las tecnologías inalámbricas han conducido al desarrollo de nuevos sistemas de comunicación y servicios multimedia. Debido al constante crecimiento del mercado automovilístico y la creciente demanda en seguridad vial ha nacido un nuevo concepto en el campo de las comunicaciones: las redes entre vehículos (VANETs). En ellas, cada vehículo actúa como router, estableciendo conexiones entre vehículos cercanos o con infraestructura en la carretera.

Las VANET estan recibiendo más atención del gobierno y de la industria automovilística debido a la amplia variedad de aplicaciones y servicios que puede ofrecer, tales como sistemas de seguridad vial, asistencia en carretera y acceso a Internet. Sin embargo, el diseño e implementación de las VANET es un area de investigación amplia y compleja, tal y como podemos percibir, sabiendo que durante los últimos años la comunidad investigadora se ha centrado en el estudio de estas redes.

Básicamente, nuestro proyecto está dividido en dos partes principales: Primeramente, hemos llevado a cabo una búsqueda relacionada con el estado de arte de las VANET hoy en día, con el objetivo de identificar los generadores de movimiento i los simuladores de redes más apropiados i recomendados en la literatura. En segundo lugar, hemos decidido utilizar el VanetMobiSim [80], como generador de movimiento debido a la alta variedad de modelos de mobilidad que que se pueden testear, y el NS2 [63] como simulador de redes por ser uno de los más utilizados per muchos autores además de su compatibilidad con el VanetMobiSim. Con el uso de estas herramientas, hemos llevado a cabo una evaluación profunda de las prestaciones de las VANET en varios escenarios reales, asignando valores diferentes a parámetros tales como el número de nodos, la velocidad y el modelo de propagación.

1. INTRODUCTION AND GOALS

Due to vehicle traffic accidents, the estimated number of deaths is about 1.2 millions people yearly worldwide and of injuries are about forty times of the previous number, without forgetting the traffic congestion that makes a huge waste of time and fuel. With the developments in wireless communications technology, the concept of Vehicular Ad hoc Network (VANETs) has taken the attention all over the world. Such network is expected to be one of the most valuable technology for improving efficiency and safety of the future transportations. Thus, several ongoing research projects supported by industry, governments and academia, have established standards for VANET networks.

In the recent years, many new projects have been proposed trying to evaluate vehicular networks. The ideal evaluation for any network (VANETs could be a choice) related to protocols and applications is to implement it in a real experiment. However this solution has many drawbacks, such as the required expensive investment that could cost implementing such experiment. An alternative evaluation that could achieve to similar results as in the real experiment is the use of simulation tools. A vehicular traffic generator and a network simulator must be coupled in order to generate complete and realistic simulations of VANETs. In this project, we have decided to use VanetMobiSim as a mobility generator, which has a wide variety of driver behaviour models. Besides, it is easily possible to simulate almost realistic road layouts taken from real maps. The network simulator used is NS2, a discrete network simulator with total compatibility with VanetMobiSim.

With this couple of simulators, we have carried out a performance evaluation of VANETs in terms of received packet ratio, average number of hops, average end-to-end delay and data throughput using several scenarios in diferent conditions.

The project consists of 9 chapters. Chapter 2 introduces the concept of Mobile Ad hoc Networks (MANET), VANET and the most known mobility generators and traffic generators used in vehicular networks. Chapter 3 describes WAVE (Wireless Access for Vehicular Environment), a new set of standards which has been developed for vehicular environment. Chapter 4 summarizes the different routing protocols available in ad hoc networks emphasizing in AODV (Ad hoc On-demand Distance Vector), the used routing protocol in this project. In Chapter 5 different propagation models are described for urban environment. Chapter 6 includes an overview of the simulation tools used in the project. Chapter 7 includes a description of the simulations settings. Chapter 8 summarizes the results obtained after a whole simulation process. Chapter 9 outlines the main conclusions and later gives a proposal for future lines of work. Finally, the project also includes several annexes which are expected to be complementary for the previous chapters.

2. INTRODUCTION TO MOBILE AND VEHICULAR NETWORKS

Over recent years, the considerable mobile services sector growth around the world was certainly the major phenomenon in the telecommunications field. Wireless technology is capable of reaching virtually every location on the surface of the earth. With such success of mobile communication demand it is hardly surprising that wireless technology has led to the development of new multimedia services and to evolution in user requirements in terms of throughput and universal mobility throughout different systems. Mobile communications are already applied to the realm of personal and business computing, making that people living habits and working ways evolve [48].

Generally there are two distinct approaches for enabling wireless mobile units to communicate each other:

Infrastructured or centralized networks*:* Wireless mobile networks have traditionally been based on the cellular concept, where all the devices are connected to a central node which is the acting agent for all communications, and relied on good infrastructure support. Typical examples of this kind of wireless networks are GSM, UMTS, WLAN, etc.

Infrastructureless*:* As to infrastructureless approach, the mobile wireless network is commonly known as a mobile ad hoc networks or MANETs. A MANET is a collection of wireless nodes that can dynamically be set up anywhere and anytime without using a pre-existing network infrastructure. It is an autonomous system in which mobile host connected by wireless links are free to move randomly and often act as routers at the same time.

The design of network protocols for MANETs is a complex and wide area of research with many challenges. Hence, during the last few years, mobile ad hoc networks have become a very popular field of study within the research community.

In the next section a more detailed description of MANETs is presented, including its main features, challenges and an introduction of two well-known applications: WSNs and VANETs. Their properties and current deployments are also discussed.

2.1. Introducing MANETs (Mobile Ad hoc NETworks)

A MANET is a collection of wireless devices than can dynamically form a network with a very simple deployment capability, paving the way for new applications which have not been able to emerge until now and offers solutions in multiple environments that have no infrastructure. These devices or nodes can move in a random way and are capable to self-organize themselves arbitrarily, collaborating in order to communications succeed. Examples of devices are laptops, PDAs, mobile phones, handhelds and wearable computers. The most outstanding features of MANETS are detailed below:

Dynamic network topology

The dynamically network topology is undoubtedly the element characterizing in MANETs. Since the nodes are mobile, the network topology may change rapidly and unpredictably and the connectivity among the terminal may vary with time. MANETs should adapt to the traffic and propagation conditions as well as the mobility patterns of the mobile networks nodes.

Autonomous terminals and self-organization

In MANETs, each mobile terminal is an autonomous node, which may function as both a host and a router and are responsible for dynamically discovering other nodes to communicate or handle the network configuration e.g. addressing, and position location issues.

Distributed operation

Since there is no background network for the central control of the network operations, the control management of the network is distributed among the devices. The nodes involved in a MANET should collaborate each other and act as a relay as needed, to implement routing and security functions.

Multi-hop routing

As delivering data packets from a source to its destination out of the direct wireless transmission range, the packets should be forwarded via one or more intermediate node.

Fluctuating link-capacity

The nature of high bit-error rates of wireless connections might be more critical in a MANET. The radio transmission rate is vulnerable to noise, fading, multiple access and interference conditions, and has less bandwidth than wired networks.

Light-weight terminal

In most cases, the MANET nodes are mobile devices with limited processing capability, small memory size and low power storage. Such devices need optimized algorithms to execute computing and communicating functions.

Scalability

Sometimes the number of devices which set up the network can increase until dozens or hundreds. Since there is no a central element which is in charge of network management, adding or rejecting nodes into the topology is a simple process.

2.2. Challenges in MANETs

Regardless of MANETs capabilities, possibilities and characteristics have risen a quickly spreading, that benefits lead to several challenges as well, which must be studied carefully. Researching in the area of mobile ad hoc networking is receiving more attention from academia, industry, and government during the last few years. Almost every aspect of the network has been explored in some level of detail although no ultimate resolution to any of the problems is found yet and there are already many open issues for research and significant contributions.

2.2.1.1. Scalability Weakness

The number of network nodes can be large and finding route to a destination also requires frequent exchange of routing control information among the nodes. Thus, the amount of update traffic can be substantial, and it is even higher when nodes with increased mobility are present.

2.2.1.2. Routing

Routing in ad hoc networks, which is quite different from traditional IP routing, is a particularly complex problem because of many factors including topology, selection of routers, locations of request initiator, resource limitations and unreliability of wireless links. A node at least needs to know the reachability information to its neighbors for determinate a packet route, while the network topology can change quite often in a MANET. Thus, routes may change while in use and become no longer valid in a very short time [67].

Since the arrival of ad hoc network concepts, many proposals have been studied, simulated and evaluated. The same proposals have led to variations, specializations to given environments and optimizations. Ad hoc routing proposals can be classified into two main categories: proactive and reactive routing. Proactive or table-driven protocols are directly inspired by routing protocols deployed in the Internet and consist of maintaining a routing table for sending data to any node in the network. Instead, ad hoc reactive routing algorithms research the vital information of a route between two nodes when a request for this route is expressed by the higher protocol layers. The protocol of this class attempt to keep the routes used and only those as up to date as possible in order to minimize the use of control messages to a minimum to save bandwidth. We can add other generally hybrid proposals to these two families, which includes both features. These categories are presented in more detailed in *Chapter 4*.

2.2.1.3. Security

Research on security in addition to routing challenges has become a primary concern to mobile ad hoc networks. Historically, network security has adopted a centralized, largely protective paradigm to satisfy aforementioned requirements. This is effective because the privileges of every node in the network are managed by dedicated machines, e.g. authentication servers. Membership in such a network allows individual nodes to operate in an open fashion because it is simplicity guaranteed that any malicious user from outside world will not be allowed access. Although these solutions have been considered very early in the evolution of ad hoc networks, attempts to adapt similar client-server solutions to a decentralized environment have largely been ineffective.

Attempts to secure ad hoc networks must be ad hoc: they must establish security without reference to centralized. Instead, security paradigms should be carried out by the cooperation of all available nodes in the network.

An implementation of an inefficient authentication protocol in ad hoc networks may lead to a vulnerability increase and network will be compromised. Attacks from malicious nodes could range from message replay, passive eavesdropping to injecting erroneous messages or liable information into routing tables in order to make network congestion and denials of service by forwarding traffic to a black hole. Hence, security solutions need to consider malicious attacks not only from outside but also from within the network. So, key management and authentication procedure are issues that must be carefully considered.

2.2.1.4. Quality of Service (QoS)

As a wired network, the flows generated by applications supported by mobile ad hoc networks have diverse characteristics such as type and the volume of exchanged information, duration of interaction to name a few examples. These flows also have different QoS requirements: bandwidth requirements for video on demand or end-to-end requirements for voice over IP services. That is why uniform packet processing is not appropriate, and QoS support which considers the different QoS requirements is vital.

In MANETs, the dynamic networks environment with continuous topology changes and the limited resources raise that problem of QoS support at different levels [48].

2.2.1.5. Energy-constrained operations

Some or all of the nodes in an ad hoc network may rely on batteries or other exhaustible means for their energy. Therefore, energy conservative networks are becoming extremely popular within the ad hoc networking research.

The goals can be achieved either by developing better batteries, or by focusing on the devices' networks interface, which is often the single largest consumer of power.

Energy efficiency at the network interface can be improved by developing transmission/reception technologies on the physical layer, but especially with specific networking algorithms. Nevertheless, energy conservation is currently being addressed in every layer of the protocol stack.

Much research has been carried out yet, however, there are still much more work to be done [48].

2.2.1.6. Interoperation

The self-organization of ad hoc networks is a challenge when two independently formed networks come physically close to each other. This is an unexplored research topic that has implications on all levels on the system design. The issue of joining two networks is not trivial: the networks may be using different synchronization, or even different MAC, routing or security protocols.

Another important issue comes into picture when we talk about all wireless networks. One of the most important aims of recent research on all wireless networks is to provide seamless integration of all types of networks. The issue raises questions on how the ad hoc network could be designed so that they are compatible with, for instance, wireless LANs, 3G and 4G cellular networks [67].

2.2.2. Applications of MANETs

With the increase of portable devices as well as progress in wireless communication, ad hoc networking is gaining importance with the number of widespread applications. Ad hoc networking can be applied anywhere where there is little or no communication infrastructure or the existing infrastructure is expensive or inconvenient to use. Ad hoc networking allows the devices to maintain connections to the network as well as easily adding and removing devices to and from the network. The set of applications for MANETs is diverse, some well-known are [67]:

2.2.2.1. Community Networking

For some business scenarios, the need for collaborative computing might be more important outside office environments than inside a building. After all, it is the case where people do need to have outside meetings to cooperate and exchange information on a given project. It is also an interesting solution for neighborhood scenarios, stadiums, museums or airports.

2.2.2.2. Crisis-management applications

That includes emergency or rescue operations, as a result of natural disasters where the entire communications infrastructure is in disarray or inoperative (Tsunamis, hurricanes). Restoring communications quickly is essential. By using ad hoc networks, an infrastructure could be set up in hours instead of days/week required for wire-line communications.

2.2.2.3. Personal Area Networking

A personal area network (PAN) is a short-range, localized network where nodes are usually associated with a given person. Bluetooth is an example of a technology aimed at supporting PANs by eliminating the need of wires between devices such as printers, cell phones and PDAs or laptop computers so on.

2.2.2.4. Military battlefield applications

MANETs networking was created for military purposes. Ad hoc networking would allow the military to take advantage of commonplace network technology to maintain an information network between the soldiers, vehicles, and military information head quarters.

Besides of these applications, two fields of study have become very interesting within the research community: wireless sensor networks and vehicular ad hoc networks. VANETs are introduced in section 2.3 whereas the special features and advantages of WSN are detailed below.

2.2.3. Wireless Sensor Networks (WSN)

In recent years, advances in wireless networking, micro-fabrication and integration and embedded microprocessors have enabled a new technological vision possible: wireless sensor networks. WSN consist of a large number of sensor nodes which collect data and interoperate each other to carry out functions involving some kind of tracking, monitoring or controlling. A sensor node is basically a device that converts a sensed attributed (such as temperature or vibrations) into a form understandable by the users. It consist of a transducer to sense a given physical quantity with a predefined precision, an embedded processor for local processing, small memory unit for storage and a wireless transceiver to transmit or receive data.

WSNs, which are considered as a special case of a MANET with reduced or no mobility, are expected to find increasing deployment in coming years, as they enable reliable monitoring and analysis of unknown and untested environments.

Development of wireless sensor networks was motivated by military applications such as battlefield surveillance and is now used in a variety of physical phenomena of interest: monitoring pedestrian or vehicular traffic in human-aware environments, report wildlife habitat conditions for environmental conservation, detect forest fires to aid rapid emergency response and monitoring industrial process or healthcare applications [89].

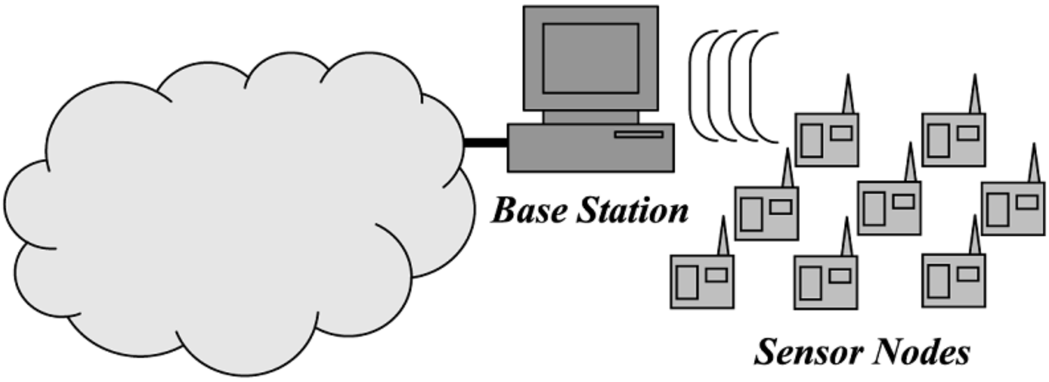


Fig. 2-1 Architecture of a WSN

2.2.3.1. Properties

The advancement in technology has made it possible to have a network of hundreds or even thousands of extremely small devices equipped with programmable computing and sensing and wireless communications capability, enhancing the reliability and the coverage area. Some of the features and challenges of WSN are as follows.

Easy of deployment

These wireless sensors can be deployed (dropped from a plane or placed in a factory) at the site of interest without any prior organization, thus reducing the installation cost and time, and also increasing the flexibility of deployment.

Fault tolerant

With macro-sensors, the failure of one node makes that area completely unmonitored until it is replaced. With wireless sensors, failure of one node does not affect the networks operation substantially as there are other adjacent nodes collecting similar data. At most, the accuracy of data collected may be somewhat reduced.

Networking and security implementation

Sensor nodes have limited computing power and therefore may not be able to run sophisticated network protocols or authentications and encryption algorithms contrary to ad hoc networks, leading to light weighted and simple versions of routing protocols and security implementations. Besides, two operational modes or states are defined for each node, awakened mode and slept mode, if the node must be active or not active, so the protocols and algorithm implementations must take into account this limitation too.

Data centric

In traditional networks, data is requested from a specific node. WSN are data centric, data is requested based on certain attributes. An attribute-based address is composed of a set of attribute-value pair query. For instance, if the query is something like temperature >35º, then only those device sensing temperature >35º need to respond and report their readings and other sensor can remain in the slept state. Once a event of interest is detected, the system should be able to configure itself so as to obtain high quality results [67].

Mobility

Since these wireless sensors are equipped with battery, they can possess limited mobility. Thus, if a region becomes unmonitored we can have the nodes rearrange themselves.

Energy conservation

Sensor nodes can use up their limited energy supply carrying out computations and transmitting information in a wireless environment. As such, energy-conserving forms of communication and computation are crucial as the node lifetime shows a strong dependence on the battery lifetime.

2.2.3.2. Current deployments and applications

Judging by the interest shown by military, academia, and the media, dozen applications do exist for sensor networks such as weather monitoring, security and tactical surveillance, detecting ambient conditions or domotic and healthy applications. A brief description of some of them is presented below:

2.2.3.2.1. Remote Ecological Micro-Sensor Network

PODS [9] is a research project undertaken at the University of Hawaii that has built a wireless network of environmental sensors to investigate species of plants will grow in one area. They deployed camouflaged sensor nodes in the Hawaii Volcanoes National Park, where two types of sensor data are collected: weather data are collected every ten minutes and image data are collected once per hour. Users employ the Internet to access the data from a server in University of Hawaii at Manoa.

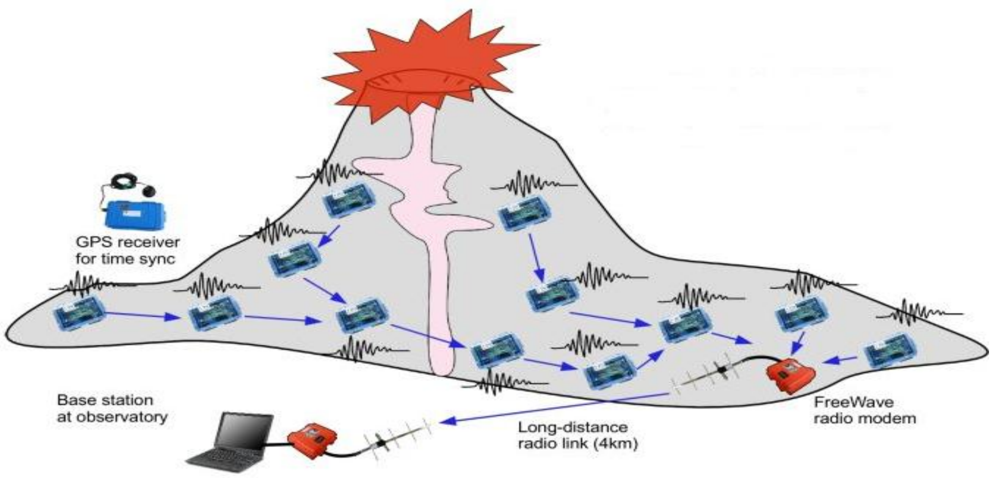


Fig. 2-2 Monitoring volcanic eruptions with a WSN [39]

2.2.3.2.2. Environment Observation and Forecasting System

The environment Observation and Forecasting System (EOFS) is a distributed system that spans large geographic areas and monitors, models and forecast physical processes such as environmental pollution or flooding. CORIE [26] is a prototype of EOFS for the Columbia River (Oregon, USA) which integrates a real-time sensor network, a data management system and advanced numerical models. Approximately thirteen stationary sensor nodes fixed to a pier are deployed across the Columbia River estuary, while one mobile sensor station drifts off-shore. The stationary are powered by a power grid, while the mobile station uses solar panel to harness solar energy. Sensor data are transmitted via wireless links towards on-shore master stations which, in turns, forward the data to a centralized server where it serves as input to a computationally physical environment model used to guide forecasting.

2.2.3.2.3. Disaster Relief Management

Novel sensor network architecture has been proposed in [17] that could be useful for major disasters including earthquakes, storms, floods, fires and terrorist attacks. The sensor nodes are deployed randomly at homes, offices, and other places prior to the disaster and data collecting nodes communicate with database server for a given sub are which are linked to a central database for continuous update. Based on the statistical data form Izmit earthquake in 1999, various performance curves are obtained to indicate required average number of active sensor nodes to detect a disaster, probability of the disaster to be within the sensing range, total number of transmitted packets and the number of sensor nodes failed due to energy depletion.

2.2.3.2.4. Health care monitoring

An example of such application is the artificial retina developed within the Smart Sensors and Integrated Microsystems (SSIM) project [74], where a retina prosthesis chip consisting of one hundred microsensors are built and implanted within the human eye allowing patients with no vision or limited vision to see at an acceptable level. Wireless communication is required to suit the need for feedback control, image identification and validation.

2.2.3.2.5. DARPA Efforts towards Wireless Sensor Networks

The Defense Advanced Research Projects Agency (DARPA) has identified networked micro sensors technology as a key application for the future. There are many interesting projects and experiments going under the DARPA SensIT (Sensor Information Technology) program which aims to develop the software for distributed micro-sensors.

Vehicle type identification is important for defense applications and an experiment was performed for two weeks by placing sensor boards in the Marine Corps Air Ground Combat Center in Twenty-nine Palms (California) for collecting acoustic data [30]. To detect presence of a vehicle, the sensor board is equipped with acoustic, seismic and passive Infra-Red sensors under program control and local processing is done to do local classification and storage.

2.3. Introducing VANETs (Vehicular Ah-hoc NETworks)

Traditional traffic management systems are based on centralized infrastructures where cameras and sensors implemented along the road collect information on density and traffic state and transmit this data to a central unit to process it and make appropriate decisions. This type of system is very costly in terms of deployment and is characterized by a long reaction time for processing and information transfer in a context where information transmission delay is vital and is extremely important this type of system. However, with the rapid development of wireless communication technologies a new decentralized architecture based on vehicle-to-vehicle communications (V2V) has created a very real interest these last few years for car manufacturers, the R&D community and telecom operators. Thus, a new concept was born: a vehicular ad hoc network (VANET), which is no more than a specific application of traditional mobile ad hoc networks (MANET).

Vehicular Ad hoc NETworks (VANETs) have recently emerged as a platform to support intelligent inter-vehicle communication to improve traffic safety. The road-constrained characteristics of these networks and the high mobility of the vehicles, their unbounded power source, and the emergence of roadside wireless infrastructures make VANETs a challenging and promising research topic.

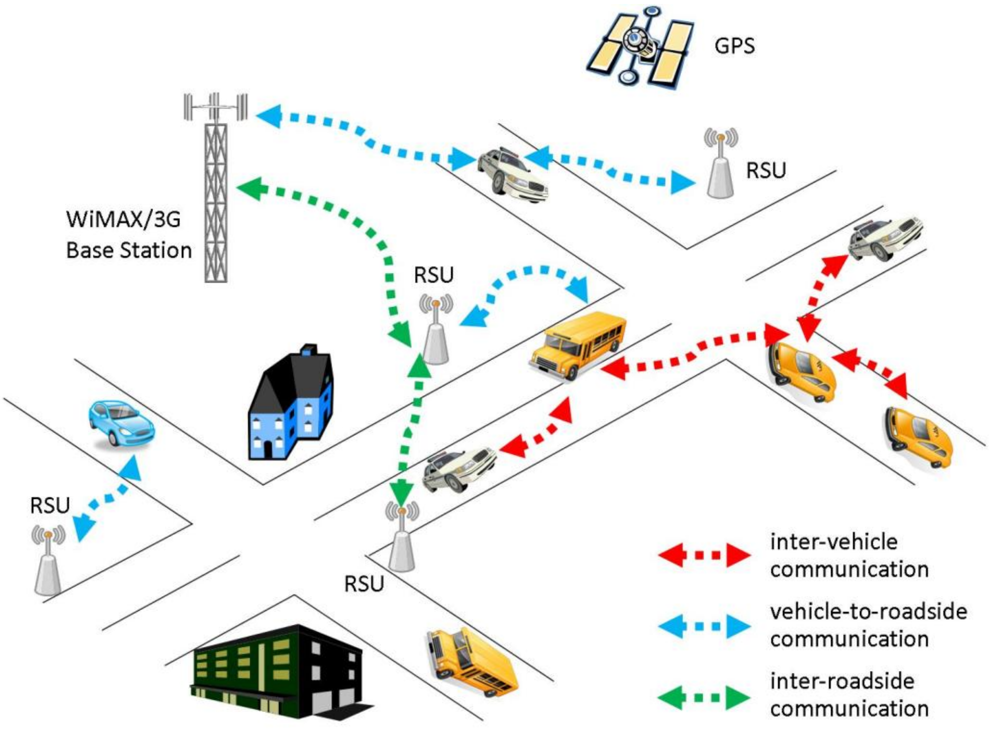


Fig. 2-3 Wireless Vehicular Networks

VANET's aim to provide our cars and roads with capabilities to make road more secure and to make our time on the road more enjoyable, enabling communications among nearby vehicles (car to car communication) as well as between vehicles and nearby fixed equipment (car to infrastructure communication). The following variety of applications is a typical example of an intelligent transportation system (ITS):

- **Safety**, in which a warning message will be broadcasted from a vehicle to its neighborhood notifying about some event such as car collision or road surface conditions in order to decrease traffic accidents rate and enhance traffic flow control. It refers to applications or systems that increase the protection of the people in the vehicle as well as the vehicle itself
- **Resource efficiency**, referring to increase traffic fluency with data such as enhanced route guidance or parking spot locator services. Better efficiency results in less congestion and lower fuel consumption, helping to minimize environmental and economic impact.
- **Infotainment and Advanced Driver Assistance Services (ADAS)**, combining *information* and *entertainment* and offering multimedia and Internet connectivity facilities for passengers.

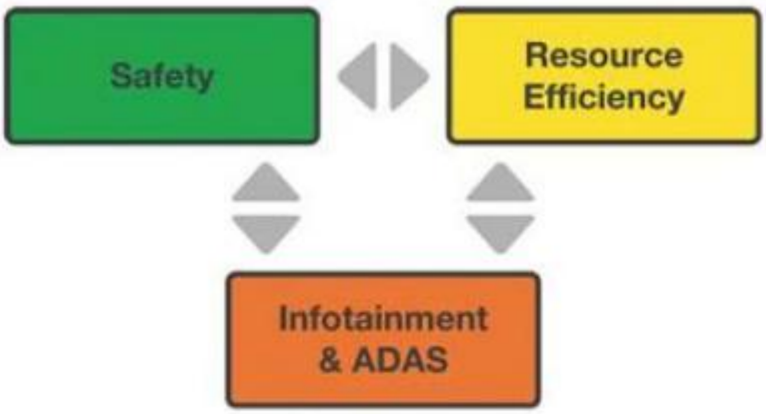


Fig. 2-4 Application domains [66]

The huge potential of car-to-car connectivity is fundamentally due to the constant growth of automotive market and the increasing demand for the car safety. Some issues relating to architecture, routing, security, performance or QoS should be investigated. It is necessary to put special attention to ensure interoperability through the standardization of protocols and interfaces in order to allow the communication between vehicles from different manufacturers.

2.3.1. Properties

As previously mentioned, a VANET represents a specific aspect of MANETs. Nevertheless, research works studied and carried out in the field of MANETs cannot be applied directly in the context of vehicular networks because of the characteristics of VANET making the application of MANET protocols and architectures inappropriate. In the following, the main properties and constraints related to the environment of vehicular ad hoc networks are presented below.

Processing, energy and communications capacity

Contrary to the context of mobile ad hoc networks where energy constraint represents one of the main challenges, vehicles in a VANET have no limit in terms of energy, have large processing capability and allow supporting several communications interfaces.

Environment and mobility model

Environments considered in ad hoc networks are often limited to open spaces or indoors. Vehicle movements are connected to road infrastructures, on highways, or within a metropolitan area. The constraints imposed by this type of environments, such as radio obstacles because of buildings, and multipath and fading effects, considerably affect the mobility model and radio transmission quality.

Type of information and diffusion

Since one of the key VANETs applications is prevention and road safety, the types of communications will focus on message broadcast from a source to several receivers. Nevertheless, the vehicles concerned by such diffusion depend on their location and their degree of implication in the event. In such situations, communications are mainly unidirectional.

Network topology and connectivity

Contrary to ad hoc networks, VANETs are characterized by very high mobility because of car speed. Thus, an element can quickly join or leave the network in a very short time, which makes topology changes frequently. Solutions must then consider this constraint where connectivity is one of the key parameter. In addition, properties inherent to VANETs, especially in terms of size, raise scaling problems and a complete revision of existing solutions is required.

Security

Data sensitivity transmitted over a VANET demonstrates a high need for security. In fact, the importance of security in this context is vital because of the critical consequences resulting from a violation or attack. In addition, with a highly dynamic environment characterized by almost instant arrivals and departures of cars, the deployment of a security solution must cope with specific configurations and constraints.

2.3.2. State of the art in VANETs

2.3.2.1. VANETs Applications

A VANET communication platform allows an enormous variety of applications aimed at administration, companies, drivers and people in the vehicle. These services will help and support topics as important as security driving, safety-related, traffic and fleet control as well as entertainment applications. Generally, from the connectivity point of view they could be divided into four groups: car-to-car traffic, car-to home, car-to-infrastructure and routing based applications.

2.3.2.1.1. Safety-related Applications

Safety-related applications are the most important kind of applications for VANETs due to its main objective: decrease of injuries and deaths due to vehicle accidents. In this context, the European Commission is making an important effort to investigate, develop and implement these services in order to come into effect as soon as possible.

Cooperative collision avoidance

This service is about helping driving by detecting possible obstacles in the road. One such application would be emergency notifications. In case of an accident or sudden hard breaking, a notification is sent to the following cars. This information could also be propagated by cars driving in the opposite direction and, thereby, conveyed to the vehicles that might run into the accident.

For a correct display of this service, it would be necessary a little installation in users' equipment which sends information about his/her position, trajectory or speed to the neighbours. Also, another system in the vehicle permanently listens to rely information from the rest of vehicles and infrastructure.

Cooperative driver assistance system

This service exploits the exchange of sensor data or other status information among cars. The basic idea is to broaden the range of perception of the driver beyond his/her field of vision and further to assist the driver with autonomous assistance applications. By transmitting this data to cars following on the same road, drivers get information about hazards, obstacles or traffic flow ahead, resulting in more efficient and safe driving. Sensors could detect a danger and warn drivers with a brief description or even the driver could detect it and, through a vocal interface, describe the danger of the situation to the rest of users.

Information could be sent to every user in the network to inform, for example, that a traffic jam has started in a certain point of the road where we are driving. On the other side, it could be necessary a geocast message if, for example, we detect an oil puddle in one exit road; is necessary to send the information only for those vehicles that will take that exit.

eCall

eCall [31] is a project of the European Commission intended to bring rapid assistance to drivers involved in a collision anywhere in the European Union. In case of crash, an eCall-equipped vehicle automatically calls the nearest emergency centre. Even if no passenger is able to speak, e.g. due to injuries, a “Minimum Set of Data” is sent, which includes the exact location of the crash site. Shortly after the accident, emergency services therefore know that there has been an accident, and where exactly. eCall cuts emergency services' response time. It goes down to 50% in the countryside and 60% in built-up areas. Annually, the quicker response will save around 2.500 lives in the European Union. The severity of injuries could be considerably reduced in 15% of cases. Drivers could also make an eCall by pushing a button in the vehicle. Witness of an accident can report it and automatically give the precise location.

2.3.2.1.2. Comfort Applications

The general aim of these applications is to improve passenger comfort and traffic efficiency. That could include nearest POI (Points of Interest) localization, current traffic or weather information and interactive communication. All kinds of applications, which may run on top of TCP/IP stack, might be applied here (online games or instant messaging). Another application is reception of data from commercial vehicles and roadside infrastructure about their businesses (’wireless advertising’), information about gas stations or enterprises which can set up stationary gateways to transmit marketing data to potential customers passing by, online help in case of breakdown, etc.

Furthermore, these services could be integrated with electronic payments, toll paying systems, etc. An important feature of comfort or commercial applications is that they should not interfere with safety applications. In this context traffic prioritizing and use of separate physical channels is a viable solution.

Optimum route calculation with real-time traffic data

This service could be used from the vehicle or from another point with an Internet connexion. The fact that, in the long term, all vehicles will be equipped with this system will make data-taking and data-publishing easier. This data, conveniently analyzed, will inform about the state of the road, prevent traffic jams, etc.

2.3.2.1.3. Applications for Administration

Vehicle identification

This service will provide a safe and fast way of information provision from vehicles without the need of stopping them. It will be necessary an appropriate legislation to allow that each vehicle stores the necessary information in an electronic format and its automatic transmission if it is required by an authorized device.

Vehicle identification service will help police in different ways: it would be possible to check if a vehicle and his/her driver have the necessary documentation or, if an infraction is detected, its report would be automatically processed, etc.

2.3.2.2. Challenges in VANETs

When deploying of a vehicular network system, several issues have to be resolved. VANET characteristics – rapid topology changes, frequent fragmentation, variable and highly dynamic scale and network density, etc. – are opening some brand new lines of investigation and challenges for the scientific community.

In what follows, we briefly mention some issues related to VANETs that need to be addressed.

2.3.2.2.1. Routing

In vehicular networks, mobility is constant. This fact causes extremely fast changes in network topology and involves the need to reconfigure the routing tables of each node. Frequent network partitioning in VANETs requires a different approach, e.g. the 'carry and forward' idea [19], where, if there is no a direct route, a packet is carried by a node until it could be forwarded to a node being closer to the destination, or the Delay Tolerant Networking.

Delay Tolerant Networking (DTN) is an approach to computer network architecture that wants to address the technical questions in heterogeneous networks, such as mobile networks, that could lack continuous network connectivity. The Delay Tolerant Network Research Group (DTNRG) [32] has defined an architecture based on a store and forward paradigm to interconnect networks, even without end-to end connectivity. Each DTN node may store packets and, when appropriate, forward them towards the destination through intermediate nodes.

2.3.2.2.2. Security

Security is an issue that needs to be carefully addressed and assessed in the design of the vehicular communication system. In a wired network, user has to access to physical wire if s/he wants to access the network's information. However, wireless communications are weak from this point of view, because they use air as the transmission medium. This problem gets worse in vehicular networks due to the non-existence of infrastructure that provides security services centralization like user authentication or packet ciphering. The issue to be addressed includes trust – vehicles must be able to trust the messages they receive–, resiliency and efficiency –e.g. real-time message authentication–.

Privacy is also considered a major issue. Anonymity must be preserved making impossible tracking a vehicle for non-trusted parties. Not taking into account privacy could result in a multiple lawsuits after the network is deployed.

IEEE 802.11p dynamically assigns MAC addresses, along with a mechanism to duplicate MAC address discovery, thus vehicles and drivers would not be traceable for the MAC address.

As we can read in the online magazine DailyTech [25], Daimler-Chrysler is putting the finishing touches on a new system, **Wireless Local Danger Warning (Willwarn)**, that uses on-board ABS (Antilock Brake System), ESP (Electronic Stability Control), EBD (Electronic Brake Distribution) and GPS (Global Positioning System) systems to monitor hazardous road conditions or broken down vehicles. The information collected is then displayed to the driver so that proper precautions can be taken to avoid or safely navigate problem areas on the road. The Willwarn system is based on IEEE 802.11a/p and made use of the communication platform developed in the German **Network on Wheels (NOW)** project [61].

2.3.2.2.3. Quality of Service (QoS)

Quality of Service in wired networks is provided by different types of resources reservation mechanisms. However, executing these mechanisms is very complex due to the special features of VANET, such as high mobility nodes and large-scale node population. Nowadays, there exist some proposals; nevertheless, the majority of them are theoretical, simulated or implemented with fewer nodes.

Tarng et al. [79] proposed a method based on the stability from the radio propagation: signal strength and path loss; Sun et al. [77] proposed a grid based protocol. The digital map is pre-set with a grid. A routing path is selected based on the traffic features, such as the intersection, the number of vehicles, and roads, in a grid. Recently, Yan et al. [88] proposed a routing selection and maintenance based on the mobility of vehicles. The main ideas are to select and maintain one routing path and one backup routing paths based on the mobility of vehicles (relative speed and direction). They reduce delay and response time in QoS terms because the retransmission caused by routing breakage is reduced and they improve the throughput in QoS terms.

2.3.2.2.4. Power Management

Power management in VANET is not concerned about energy efficiency, but rather about the transmission power – when too high, the on-going transmission could disrupt another transmission at a distant node due to interferences –. Thus the denser the network is, the lesser transmission power should be used. Several algorithms could be employed here, e.g. in [14] the power is adjusted to keep the number of neighbours within the maximum and minimum threshold. On the other hand, [51] concentrates on improving the 1-hop broadcast coverage by transmission power adjustments.

2.3.2.3. Current projects and activities in VANETs

2.3.2.3.1. Standardization process and research projects initiatives in VANETs

This section briefly explains the main progress and purposes of the standardization process and research projects initiatives. It is foreseen that these solutions will finally converge, leading to a common, worldwide VANET platform.

In **Europe**, several projects are held, joining partners from the industry, governmental agencies and academia. Within the Framework Programme 6 of the European Union, four integrated projects were started in areas that touch the field of VANET: COOPERS, CVIS, SAFESPOT and PReVENT:

- The project **Co-operative Systems for Intelligent Road Safety (COOPERS, 2006–10)** [23] focuses on innovative telematics applications for cooperative traffic management. From a communication perspective, it therefore primarily addresses vehicle-to-roadside communications and makes use of CALM (Communication Architecture for Land Mobile environment) standards like the CALM infrared communication interface. CALM is the ISO TC 204 (ITS) Working Group 16 (Communication) on 'Continuous Air interface for Long and Medium distance'. CALM aims to support continuous communications for vehicles by making use of various media and communication interfaces.
- The project **Co-operative Vehicle-Infrastructure Systems (CVIS, 2006–10)** [24] is a major European research and development project financed by the European Commission aiming to design, develop and test the technologies needed to allow cars to communicate with each other and with the nearby roadside infrastructure. Based on such real-time road and traffic information and although there are no ultimate results because of the project is still in progress, many novel applications have been produced as well as four new services: COMM, allowing car-2-X coomunications, POMA, providing positioning system, COMO, providing monitoring services and FOAM, linking vehicles to infrastructure. The consequence will be increased road safety and efficiency, and reduced environmental impact. The CVIS main challenges are:
    - To create an unified technical solution allowing all vehicles and infrastructure elements to communicate with each other.
    - To enable a wide range of potential cooperative services to run on an open application framework in the vehicle and roadside equipment.
    - To define and validate an open architecture and system concept for a number of cooperative system applications, and develop common core components to support cooperation models in real-life applications and services for drivers, operators, industry and other key stakeholders.
- The project **SAFESPOT (2006–10)** [72] aims to design cooperative systems for road safety based on vehicle-to-vehicle and vehicle-to-infrastructure communication.
- The project **PReVENT (2006–08)** [68] addressed to development of preventive safety applications and technologies. Within the PReVENT Integrated Project, the subproject Willwarn, wich we have mentioned before, focused on the topic of vehicle-to-vehicle and vehicle-to-infrastructure commu-nication.

The **Car2Car Communication Consortium** [16] is a non-profit organization initiated by European six vehicle manufacturers (General Motors, Ford Motor, Honda, Toyota, BMW and Mercedes-Benz) in 2004, pushing for further increase of road traffic safety. Its mission it to create an open European industry standard for Car2Car communication systems based on wireless LAN components and guarantee European-wide inter-vehicle operability. That includes proposing of realistic deployment strategies and business models to speed-up the market penetration.

The Car2Car consortium has established its objective of improving road safety and efficiently managing traffic using inter-vehicular systems. The Car2Car consortium together with IEEE have developed IEEE 802.11p standard, which defines enhancements to 802.11 required to support Intelligent Transportations Systems (ITS) applications. Car2Car main missions were as follows:

- The creation of an open European standard for V2V communications based on wireless LAN components.
- Developing V2V system prototype demonstrator for road safety applications.
- Promoting tre allocations of a free exclusive band for Car2Car applications in Europe.
- Developing deployment strategies and economic models for market penetration.

The **FleetNet** (Internet on the road) project [33] is a German project introduced by a consortium of six manufactures (DaimlerChrysler AG, FHI FOKUS, NEC Europe Ltd., Robert Bosch GmbH, Siemens AG, and Temic Speech Dialog Systems GmbH) and the universities of Braunschweig, Hamburg-Harburg and Hannover. FleetNet’s objective was to develop a communications platform for vehicle networks, to implement a demonstrator, and to standardize the proposed solutions in order to ensure better security and comfort for driver and passengers. The FleetNet architecture is based on a routing mechanism based on a system of location and navigation, and also considers vehicle to infrastructure communications in order to provide Internet access service.

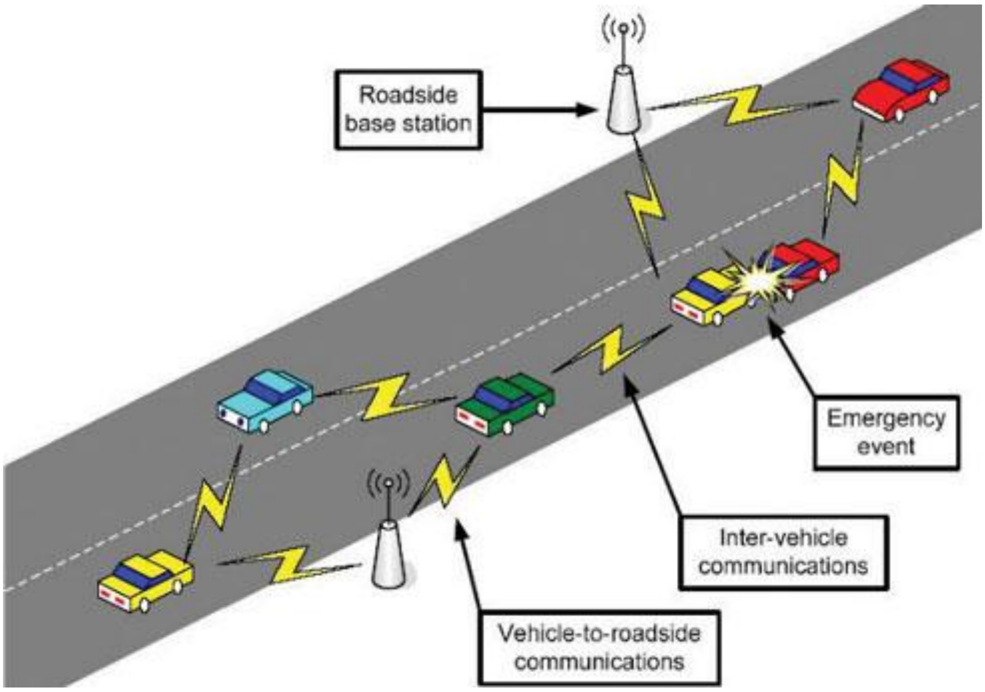


Fig. 2-5 Architecture on VANETs communication [13]

The **NOW** (Network on Wheels) project [61] is a German project from the Federal education and research government, founded by automobile manufacturers, telecommunications operators and academia. NOW is the successor of FleetNet Project and supports and strongly cooperates with the Car2Car consortium. One of NOW's main objectives is the implementation of communication protocols and data security algorithms in vehicular network. Considering the wireless 802.11 technologies and location-based routing in a V2V or vehicle to infrastructure communication context, the goal is to implement a system of reference and to contribute to the standardization of such a solution in Europe in collaboration with the Car2Car consortium.

GST (Global System of Telematics) [36] is an EU-funded Integrated Project that is creating an open and standardized end-to-end architecture for automotive telematics services. Participants were major car manufacturer and major Telecom players, altogether around 60 companies. The purpose of GST is to create an environment in which innovative telematics services can be developed and delivered cost-effectively and hence to increase the range of economic telematics services available to manufacturers and consumers. GST has introduced seven service-oriented sub-projects that seek to contribute to achieving the main targets: rescue, safety, payment, safety channel or extended floating car data.

In **Japan**, a standard for vehicle-to-infrastructure communication was published in 2001 and denoted as 'Dedicated Short-Range Communication System' (ARIB 2001). The specified system operates in the 5.8 GHz frequency band, is based on time division multiple access and targets a range of about 30 m. The primary use of the system was seen in electronic toll collection but the system was generalized to support various other services. In 2008, more than 20 million on-board units for electronic toll collection were deployed in Japan. Based on the success on this 5.8 GHz DSRC system and on infrared- based vehicle-to-infrastructure communication, various ITS projects and activities are currently joining forces to demonstrate and enhance vehicle-to-infrastructure and vehicle-to-vehicle communication under the umbrella of Japan's national ITS Safety 2010 initiative.

In the **USA**, there are several industry/government projects on-going. **Vehicle Infrastructure Integration (VII)**, which has been rebranded as **IntelliDrive** [41]**,** has recently completed a large proof of concept demonstration. The majority of this testing environment was implemented in Detroit. This system comprised 55 Road Side Equipment (RSE) stations within 45 square miles and employed 27 vehicles. Seven applications were developed and tested:

- In-Vehicle Signage: RSEs trigger displays of advisory messages within the vehicle.
- Probe Data Collection: Vehicles provide historical data on their location/state and share with the RSE, which is then centrally compiled and analyzed.
- Electronic Payments – Tolling.
- Electronic Payments – Parking.
- Traveler Information / Off-Board Navigation.
- Heartbeat: RSEs collect periodic status messages from vehicles including vehicle speed and location.
- Traffic Signal Indication: Broadcasts traffic light state.

Integrated Vehicle-Based Safety Systems (IVBSS) project [42] explores human-machine interface issues when several safety applications, with potentially overlapping or contradictory advisories, are operated simultaneously. The **Cooperative Intersection Collision Avoidance System (CICAS)** project [22], had three components: a Violation Warning project (demonstrated in Michigan), a Stop Sign Assist project (demonstrated in Minnesota), and a Signalized Left Turn Assist project (demonstrated in California).

2.3.2.3.2. Vehicular Mobility Models

Due to the prohibitive cost of deploying and implementing such a system in real world, most research in VANET relies on simulations for evaluation. A key component for VANET simulations is a realistic vehicular mobility model. Mobility models represent real world scenarios for vehicular ad hoc networks and play a vital role in the performance evaluation of routing protocols. More research focus is now on the development of realistic mobility models for VANETs. A number of mobility models have been presented and their impact on the performance on the routing protocols has been tested. To get accurate results, the model should be as realistic as possible, and involve road maps with all the constraints and facilities related to the vehicular movement. Below, we present some new mobility models specifically proposed for VANETS:

The **Integrated Mobility Model (IMM)** [3] is an integration of Manhattan mobility model, freeway mobility model, stop sign model, traffic signs model and some other characteristics like stationary nodes. The advantage of IMM is that it provides a more detailed scenario for the simulation of VANETs by representing both the rural and urban area which is clear from the simulation results. After simulate with three different routing protocols (AODV, DSR and OLSR) and compare results, they obtained that OLSR and AODV performs better than DSR in a more stressed urban scenario. The future dimensions of this work are that they will add more realistic parameters to IMM and will enhance it for VANETs simulations for more comprehensive results. The code, available at [3], has been developed by M. Alam, M. Sher and S. A. Husain at the University of Islamabad, Pakistan.

MEtropolitan TAxis (META) [40] is a mobility model, proposed in the Shanghai Jiao Tong University, in collaboration with the State University of New York at Buffalo, that can be used to generate synthetics trace for the movement of taxis in an urban area. In order to characterize the regularity of taxi movement, the authors designed three model parameters: turn probability, road section speed and travel pattern. Through different validation results, they show that META has a good approximation to a real scenario which in turn shows the effectiveness of these parameters. Based on the validation, synthetic traces can be generated; they are similar to the reality using such parameters. Since these parameters are easier to be obtained than the real trace, the META model can be used to replace the high cost real trace on some extent in other VANET researches.

MObility model generator for VEhicular networks (MOVE) [50] is a tool that allows users to rapidly generate realistic mobility models for VANET simulations. The output of MOVE is a realistic mobility model and can be immediately used by popular network simulators. The authors warn that if simple mobility models are used for evaluation of VANET, the results might not be as close to reality as expected, so they show that the details of a mobility model such as the existence of traffic lights, driver route choice and car overtaking behaviour can have a significant impact on the simulation results.

Bonnmotion is a Java software, available at [11], which creates and analyses mobility scenarios. It is developed within the Communication Systems group at the Institute of Computer Science 4 of the University of Bonn, Germany, where it serves as a tool for the investigation of mobile ad hoc network characteristics. The scenarios can also be exported for the network simulators NS2, GloMoSim/QualNet, COOJA, MiXiM, ONE and NCTUns, using a conversion script developed in the UPC (Polytechnic University of Catalonia) by Guillermo Diaz. Several mobility models are supported (Random Waypoint, Random Walk, Gauss-Markov, Manhattan Grid, etc.).

CityMob [55] is a mobility pattern generator for VANETs, designed to be used with the NS2 simulator developed in the Polytechnic University of Valencia. CityMob can generate traces for VANETs scenarios using three different mobility models: Simple, Manhattan, and Downtown.

2.3.2.3.3. Other research areas in VANETs

Apart from mobility models, VANET research covers other areas, as transportation systems, routing protocols and new infrastructures.

Intelligent transportation systems (ITSs) [2] cover different technologies as wireless communications, sensor networks, voice and data communication, real-time driving-assistant systems, etc. The interconnection of different networks, even in the case of the Internet, is one of the main difficulties that is delaying the wide spread of vehicular networks. **Location-Aided Gateway Advertisement and Discovery** (**LAGAD)** [2] is a scalable hybrid adaptive protocol that aims to reduce congestion on single channels through a channel diversity mechanism that uses multiple channels and multiple interfaces for the propagation of gateway requests and replies.

Bypass-AODV (Bypass Ad hoc On-demand Distance Vector) [4] is a new optimization of the AODV routing protocol for mobile ad-hoc networks proposed as a local recovery mechanism to enhance its performance. It uses a specific strategy of cross-layer MAC-notification to identify mobility-related packet loss, and then setup up a bypass between the node at which the route failure occurred and this node's previous successor via an alternative node. Simulation results show that Bypass-AODV is insensitive to any random mobility model used and has a clear performance improvement compared to AODV. It has a comparable performance under group mobility model compared to AODV. Currently, Bypass-AODV is not suitable for VANET applications because the movement of vehicles is constrained by the layouts of the roads. As a future work, Bypass-AODV needs more improvement in order to handle VANET applications.

With the help of VANETs, vehicles on the road can form wireless ad hoc mesh networks (VMeshs) [52]. These meshes can help to retain certain transient information in a specific region for a period of time, by cooperatively passing the information among themselves without any infrastructure help. Nevertheless, there is a VANET storage problem. Studies show that the transmission range has high impact on the storage lifetime for one-way highway traffic, and the size of the region in which we want the information stored has high impact for two-way highway traffic.

A new two-tier architecture called **Mobile Infrastructure Based VANET (MI-VANET)** [53] has been recently proposed. In this architecture, the buses constitute a mobile backbone for data delivery while the low tier is composed of ordinary cars and passengers. MI-VANET will not only bring the benefit that ordinary cars do not have to forward packets for other nodes, but also improve the network connectivity. They are currently working and studying to demonstrate that MI-VANET with MIRT performs much better than GPSR and VANET MIRT in terms of packet delivery ratio and throughput.

2.4. Vehicular Traffic Models in VANETs

Transportation and traffic research area classifies traffic models according to the granularity with which traffic flows are examined. Macroscopic models model traffic at a large scale, treating traffic like a liquid applying hydrodynamic flow theory to vehicle behavior. The simulation in a macroscopic model takes place on a section-by-section basis rather than by tracking individual vehicles. However, they do not have the ability to analyze transportation improvements in as much details as the microscopic models.

Mesoscopic models combine the properties of both microscopic and macroscopic simulation models. As in microscopic models, the mesosocopic models unit of traffic flow is the individual vehicle. However, their movements follow the approach of the macroscopic model and are governed by the average speed on the travel link, so movements do not consider individual dynamic vehicle speed and volume relationships.

Microscopic simulations, which model the behavior of single vehicles and the interactions between them, are the most appropriate mobility models for simulating VANETs. Transportation and traffic science has developed a number of microsimulation models, each taking a dedicated approach ranging from coarse to fine grain.

When dealing with vehicular mobility modeling, some authors have distinguished between macro-mobility and micro-mobility. For macro-mobility they refer to all the macroscopic aspects which influence vehicular traffic, for example: the road topology, constrained car movements, the per-road speed limits. Micro-mobility refers instead to the drivers' individual behavior when interacting with other drivers or with the road infrastructure, for instance, traveling speed under different traffic conditions, acceleration, deceleration and overtaking criteria. For a trustworthy VANET simulation that both macro-mobility and micro-mobility descriptions are jointly considered when modeling vehicular movements.

There are some models proposed in the general MANET context usable in VANET. The criteria of applicability considered are the employment of road maps, and limiting the nodes movements into the roads instead of moving in a wide area. The considered parameters differ from a model to another. For instance, some models use traffic control mechanisms at intersections, and some others just assume continuous movement at these points. Some assume roads to be a single-lane, but some others support multi-lanes roads. Some define the security distance, while others just ignore this parameter. In the following is going to be described the main features of the most spread vehicular traffic models.

In the following, the main features of the most spread vehicular traffic models are described.

2.4.1. Freeway model

Freeway [7] is a generated-map-based model, defined in the simulation area, represented by a generated map, includes many freeways, each side of which is composed of many lanes. No urban roads, thus no intersections are considered in this model. At the beginning of the simulation the nodes are randomly placed on the lanes, and they move using history-based speeds, where the speed of each vehicle smoothly changes following a random acceleration. In addition to the realism related to the acceleration and the history-based speed, the model defines a security distance that should be maintained between two subsequent vehicles in a lane. If the distance between two vehicles is less than this required distance, the second one decelerates to enable the forward vehicle moving away. The change of lanes is not allowed in this model. The vehicle moves on the lane it is placed in until reaching the simulation area limit, and then it is placed again randomly in another position and repeats the process.

2.4.2. Manhattan model

Manhattan model [7] is also a generated-map-based model introduced to simulate an urban environment. Before starting a simulation a map with vertical and horizontal roads are generated. Each road latter includes two lanes, allowing the movement in the two directions (north/south for the vertical roads and east/west for the horizontal ones). At the beginning of a simulation, vehicles are randomly put on the roads. Afterwards, they move continuously according to history-based speeds (exactly like Freeway). When reaching a crossroads, the vehicle randomly chooses a direction to follow. That is, continuing straight forward, turning left or turning right. The probability of each decision is set by the authors respectively to 0.5, 0.25 and 0.25. The security distance is also used in this model and nodes follow the same strategy as in the freeway model to maintain this distance. But contrary to the previous model, a vehicle can change a lane at a crossroads. Nonetheless, there is no control mechanism at these points (crossroads), where nodes continue their movements without stopping.

2.4.3. City Section Mobility (CSM)

CSM [27] can be viewed as a hybrid model between Random Waypoint Model (RWP), in which mobile nodes move randomly and freely without restrictions, and Manhattan as it introduces the principle of RWP, especially the pause-time and random selection destination, within a generated-map-based urban area. At each step of the vehicle's movement a random point is selected from the generated road map, towards which it moves following the shortest path. After reaching that destination, it remains there for a pause-time, and then repeats the process. The speed of nodes is constrained by the security distance, along with the maximum speed limit of the road.

2.4.4. Stop Sign Model (SSM)

Contrary to the previous models, SSM [54] integrates a traffic control mechanism. In every crossroads, a stop signal is set, which obliges vehicles to slow down and make a pause there. This model is based on real maps of the TIGER/Lines database [5], but all roads are assigned a single lane in each direction. A vehicle should never overtake its successor (like in all the models presented before) and should tune its speed to keep the security distance. If many vehicles arrive at an intersection at the same time, they make a queue, and each one waits for its successor to traverse the crossroads. This results in gathering of nodes, and hugely affects the network connectivity as well as the mobility (average speeds). According to the authors, the problem with this model is the unrealistic disposition of the spot signals, since it is impossible to find a region with spot signals at each intersection, therefore, they improved SSM and they proposed TSM.

2.4.5. Traffic Sign Model (TSM)

In TSM model [54], stop signals are replaced by traffic lights. A vehicle stops at crossroads if it encounters a red stoplight; otherwise it continues its movement. When the first vehicle reaches the intersection, the light is randomly turned red with probability *p* (thus turned green with probability 1-*p*). If it turns red, then it remains so for a random delay (pause-time) forcing the vehicle to stop, as well as the ones behind it. After the delay, it turns red, and then the nodes traverse the crossroads one after the other until the queue is empty. When the next vehicle arrives at the crossroads the process is repeated.

2.4.6. STRAW (Street Random Waypoint)

STRAW [20] is also a model using real maps of TIGER/Line [54]. Like the other models, except freeway, roads include one lane in each direction and it is divided into segments. The model is basically composed of three modules: intra-segment mobility manager, inter-segment mobility manager, and finally the rout management and execution module. At the beginning of the simulation the nodes are placed randomly one behind the other, they move using the car following and try to accelerate until reaching the maximum speed of the segment. The first module manages this movement until reaching an intersection. The security distance is maintained, but the overtaking is not allowed. At crossroads the vehicles always slow down, even when they change a segment and turn without a full stop, which is realistic. The second module defines the traffic control mechanism including both stop signals and traffic lights, which are put on crossroads according to the class of the intersected roads. In addition to this usual control form, the module makes sure that the next segment to take contains enough available space before moving the vehicle towards it. If it is fully busy, the vehicle waits at the crossroads (at the end of the first segment). The last module selects the routes to be taken by each vehicle during the simulation. In the first one the direction is randomly selected at each intersection, for example, when reaching an intersection, the vehicle randomly decides whether to continue straight forward or to turn and change the road. On the other hand in the second approach a destination is selected toward which the vehicle moves using the shortest path.

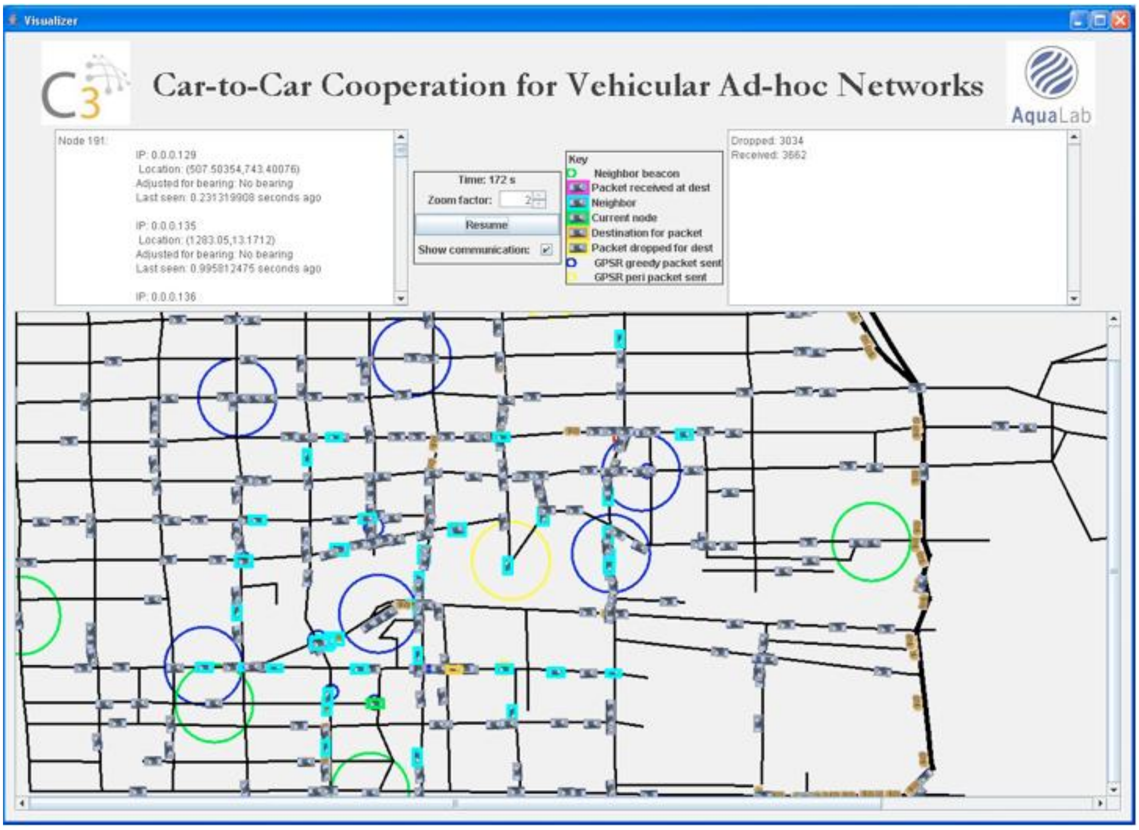


Fig. 2-6 Screenshot of STRAW with the enhanced visualization tool

2.4.7. MOVE (MObility model generator for VEhicular network)

MOVE [46] is a VANETs mobility model that uses the compiler SUMO [76], which is a realistic vehicular traffics simulation model. SUMO is an open source application implemented with java that integrates many realistic parameters such as realistic accelerations; the usage of real maps reflecting several types of roads (with multiple lanes), and traffic lights defining priorities between vehicles. Basically, MOVE is composed of two components; the road map editor and the vehicle movement editor. The former serves to manually and randomly generate a road map, either from TIGER/Line [5] files or Google earth files, whereas the latter allows specifying the properties of each vehicle like the maximum speed, the acceleration, the probability of turning at crossroads and the path to take. The information collected by the two editors is sent to the SUMO compiler.

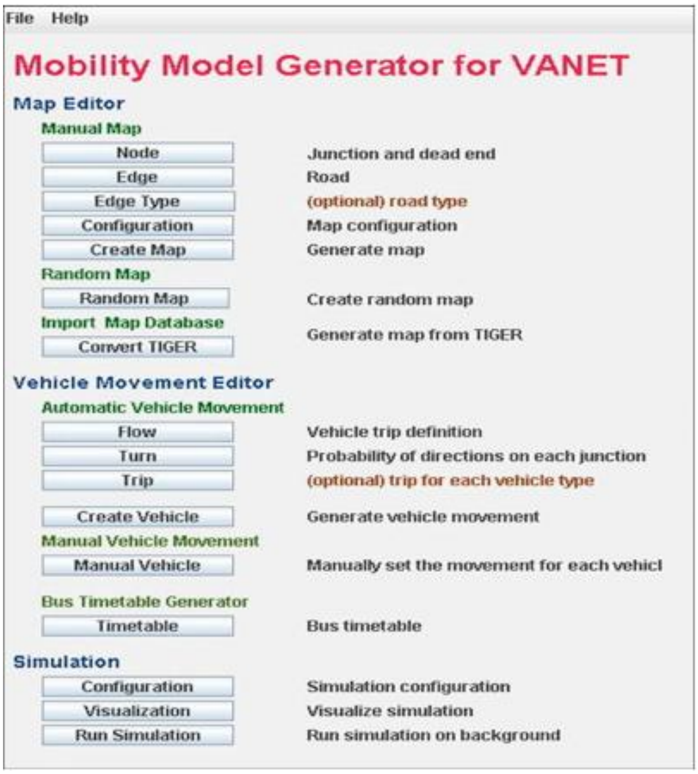


Fig. 2-7 Screenshot of MOVE

2.4.8. BonnMotion

BonnMotion tool [11] implements several random mobility models, plus the Manhattan model. While the important tool includes the Car Following Model which is a basic car-to-car inter-distance control schema, the BonnMotion does not consider any micro-mobility. When related to the framework, we can easily see that the structure of both tools is definitely too simple to represent realistic motions, as they only model basic motion constraints and hardly no micro-mobility.

2.4.9. VanetMobiSim

VanetMobiSim [37] is an extension of the CANU Mobility Simulation Environment (CanuMobiSim) [15], which focuses on vehicular mobility, and features realistic automotive motion models at both macroscopic and microscopic levels. At the macroscopic level, VanetMobiSim can import maps from TIGER/Line database, or randomly generate them. VanetMobiSim adds support for multi-lane roads, separate directional flows, differentiated speed constraints and traffic signs at intersections. At the microscopic level, it manages lane changes and vehicle accelerations and decelerations, providing realistic car-to-car and car-to-infrastructure interactions. VanetMobiSim is the mobility generator used in this project. Further description for VanetMobiSim is presented in *Chapter 6*.

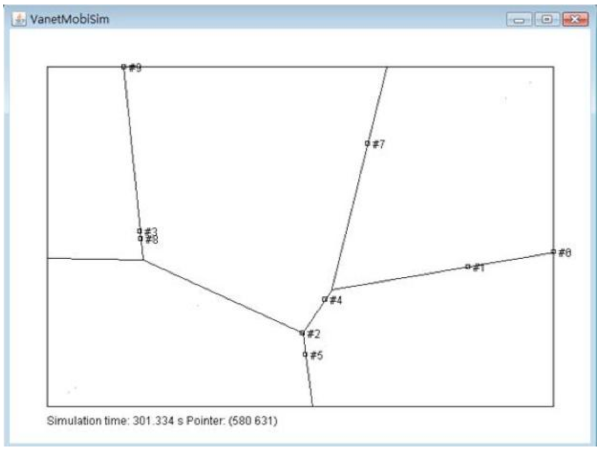


Fig. 2-8 Screenshot of VanetMobiSim

2.4.10. SUMO (Simulation of Urban Mobility)

SUMO [47] is an open source, highly portable, microscopic road traffic simulation package designed to handle large road networks. Its main features include collision free vehicle movement, different vehicle types, single-vehicle routing, multi-lane streets with lane changing, junction-based right-of-way rules, hierarchy of junction types, an openGL graphical user interface (GUI), and dynamic routing. SUMO can manage large environments, i.e., 10 000 streets. SUMO can simulate traffic in different locations of the globe. However, since SUMO is a pure traffic generator, its generated traces cannot be directly used by the available network simulators, which is a serious shortcoming.

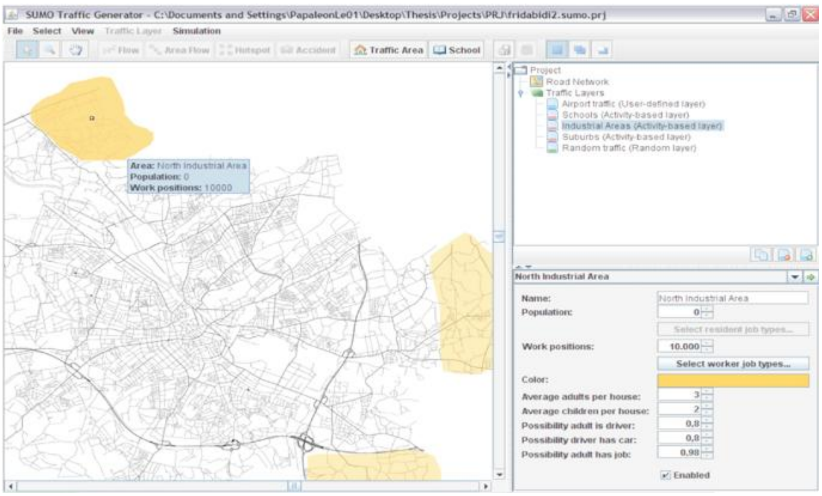


Fig. 2-9 Screenshot of SUMO

2.4.11. FreeSim

FreeSim [34] is a fully customizable macroscopic and microscopic free-flow traffic simulator that allows for multiple freeway systems to be easily represented and loaded into the simulator as a graph data structure with edge weights determined by the current speeds.

Traffic and graph algorithms can be created and executed for the entire network or for individual vehicles or nodes, and the traffic data used by the simulator can be user generated or be converted from real-time data gathered by a transportation organization. Vehicles in FreeSim can communicate with the system monitoring the traffic on the freeways, which makes FreeSim ideal for Intelligent Transportation System (ITS) simulation. FreeSim is licensed under the GNU General Public License, and the source code is available freely for download.

2.4.12. CityMob

CityMob [55] is a NS2 compatible mobility model generator proposed for use in VANETs. Citymob implements three different mobility models: (a) Simple Model (SM), (b) Manhattan Model (MM), and (c) realistic Downtown Model (DM).

Last models are programs witch includes the models explained at the beginning of the chapter. To summarize all the information there is a table witch shows the main characteristics of the last seven models, which are the most used.

	VanetMobiSim	SUMO	MOVE	STRAW	Freesim	CityMob
Maps						
Real	X	X	X	X	X	
User-defined	X	X	X			
Random	X	X	X			
Manhattan						X
Mobility						
Random Waypoint	X	X	X			X
STRAW		X	X	X		
Manhattan		X	X			X
Downtown						X
Traffic Models						
Multilane roads	X	X	X	X		X
Lane changing	X	X	X	X		X
Overtaking criteria	X					
Colision free movement		X	X			X
Different vehicles type		X	X			X
Traces						
ns-2 trace support	X		X			X
NCTUns trace support						
Import different formats	X	X	X	X		

Table 2-1 Comparison of Vehicular Traffic Models

As it can be seen none of these seven models have NCTUns compatibility but some have NS2 trace support, which it is the most usual compatibility with all VANET simulators.

3. WIRELESS ACCESS FOR VEHICULAR ENVIRONMENT. IEEE 802.11P

3.1. Introduction

Vehicular Ad-hoc Network (VANET) is a special type of Intelligent Transportation System (ITS) which is a specialization of mobile ad-hoc networks focused on vehicular environments. These networks have no fixed infrastructure and rely on the network nodes to provide networks functionality. In last years much effort is being carried out in order to research this network paradigm.

Because of the nature of vehicular environment, VANET communications must face additional challenges that are not present in other mobile communication technologies. There are two major challenges: speed and loss of Line Of Sight (LOS). Mobile nodes are vehicles moving in predefined road which depends on the road structure and traffic regulation at high speeds, which in relative measures are well above 120Km/h. For example, when two vehicles moving in opposite directions in a highway communicate with one another, the communication module may face relative speed of 240Km/h. On the other hand, some situations result in non-LOS. The lack of LOS produces degradation in the quality of the communications, which could lead to burst errors or even to complete loss of the communication. Moreover, in urban scenarios, the buildings surrounding the roads produce signal reflections that harm the communication quality because they result in burst errors that may lead to packet losses.

Although most of the essential features of the MANET are in VANET with some behavioral changes so that the routing protocols and IEEE standards used in MANET are also applied in VANET environment, it is known that IEEE 802.11 standard is not well suited for VANET environment and new technologies are being standardized for vehicular communications. An IEEE working group has developed a new set of standards which is designed for VANET: the Wireless Access in Vehicular Network (WAVE) whose outstanding component is the IEEE 802.11p amendment [71].

However, while new vehicular networking concepts are developed, other mobile and wireless technologies shall be used for VANET communications. A widespread example of these technologies is IEEE 802.11b, which is commonly known as belonging to the WIFI family of standards. Although IEEE 802.11b was initially designed for low-mobility indoor wireless scenarios, nowadays it is one of the most commonly used technologies for the experimentation of VANET communications [35]. 802.11b has been selected because has shown a more stable communications in initial test performed compared with 802.11g that showed higher transmission rates but also lower ranges [86]. Although 802.11a is similar in someway to the 802.11p, we considered that due to 802.11a works in the 5GHz IMS band, in non-LOS scenarios its performance would be worse because of its lower penetration rate. The most remarkable features of IEEE 802.11 standards used for experimentation of VANET is shown in Table.3-1.

Parameter	***802.11b***	***802.11a***	***802.11p***
Available data rate	*1-2-5.5-11 Mbps*	*6-9-12-18-24-36-48-52 Mbps*	*3-4.5-6-9-12-18-24-27 Mbps*
Allocated spectrum	*2.4-2.4835 Ghz ISM band*	*5.125-5.850 Ghz ISM band*	*5.850-5.925 Ghz DSRC band*
Modulation encoding	*DSSS*	*OFDM*	*OFDM*
Modulation mode	*BPSK-QPSK*	*BPSK-QPSK 16QAM-64QAM*	*BPSK-QPSK 16QAM-64QAM*
Communication range	*< 150m*	*< 100m*	*< 1000m*
Transmission power for mobile (maximum)	*760 mW (US) 2 W EIRP (EU)*	*100mW*	*100mW*
Suitable for mobility	*Low*	*Low*	*High*

Table 3-1 Comparison of the used IEEE 802.11 standards for VANET environment

3.2. Wireless Access For Vehicular Environment (WAVE)

In 1999, the U.S. Federal Communication Commision (FCC) allocated in the USA a 75MHz spectrum in the 5.9GHz (5.855 – 5.925) band. This band was called Dedicated Short Range Communication (DSRC) band and its use was exclusively for vehicle-to-vehicle and vehicle-to-infrastructure communication purposes. In the following years Government, industry and academia carried out efforts to develop and deploy communications systems and protocols operating in the DSRC band, known as Wireless Access for Vehicular Environment (WAVE) systems. To simplify interoperability between different automobile manufacturers for WAVE-based ITS applications, the IEEE decide to standardize the entire protocol suite.

Today WAVE is a set of standards and protocols which goal is to facilitate the provision of wireless access in vehicular environment, whose primary goal is develop applications in order to achieve public safety and trafic flow improvement. This set comprehends the IEEE 802.11p and IEEE 1609.1-4 standards.

3.2.1. Architecture Overview

WAVE provides a communication protocol suite optimized for the vehicular environment developed by IEEE, employing both customized and general-purpose elements. The components of the system, as defined in the standards, are:

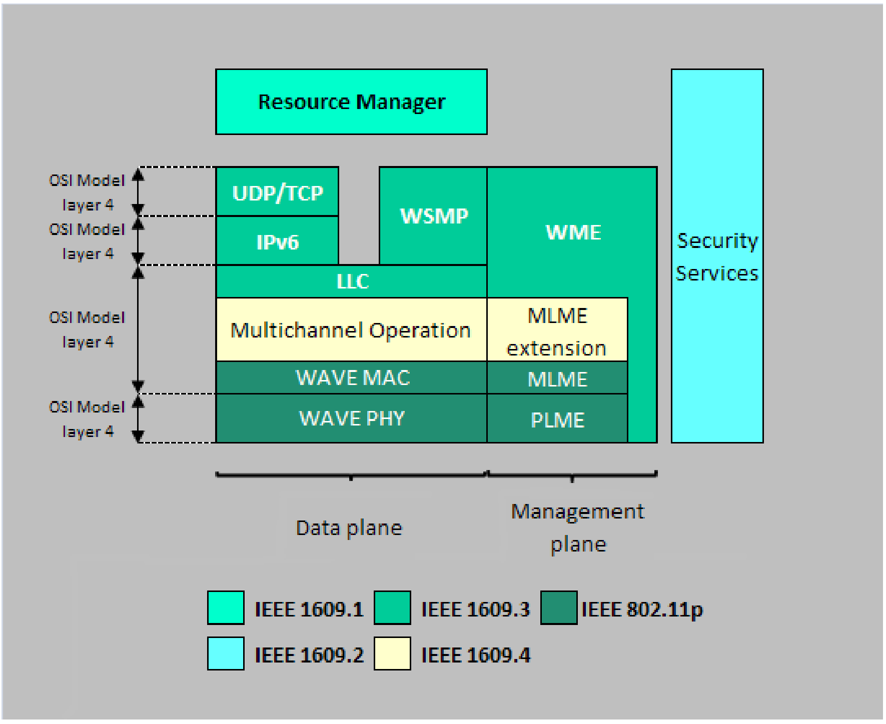


Fig. 3-1 WAVE architecture overview

- An MAC and PHY layer derived from IEEE 802.11, adressed in IEEE 802.11p. The 802.11p MAC layer is designed to be PHY independent so that both MAC and PHY layers conceptually include management entities (MLME and PLME) for this purpose.
- A multi-channel operation layer in IEEE 1609.4, where the enhancements to 802.11p MAC to support WAVE are specified.
- A new transport/network layer protocol, IEEE 1609.3, where services, operating at the network and transport layers, in support of wireless connectivity are specified. This includes the management of the WAVE BSS (WBSS). The standard call them WAVE networking services, which can be functionally divided in data-plane services, whose function is to carry trafic, and management-plane services, whose functions are system configuration and maintenance.
- Security issues specified in IEEE 1609.2, where the security services for WAVE networking stack are defined. These services provide confidentiality, authenticity, integrity and anonymity.
- An application protocol called IEEE 1609.1, which deals with resource management, e.g. describing key components of WAVE system architecture and defining data flows, command message and data sotrage formats.

3.2.2. IEEE 802.11p

The 802.11p amendment modifies the IEEE 802.11 standard to add support for wireless local area networks (WLANs) in a vehicular environment [82]. The main enhancements are:

- Short latency
- Ranges from 1 meter up to 1000 meters
- WAVE devices installed in vehicles operating at speeds up to 200 km/h
- Extreme multipath environments with many reflections and Doppler shift
- Nature of the automotive applications to be supported (e.g. reliable broadcast)

IEEE 802.11p is intended to make the minimum necessary changes to IEEE 802.11a PHY layer so that WAVE devices can communicate effectively among vehicular environments. In the following, PHY and MAC layer of IEEE 802.11p are discussed.

3.2.2.1. PHY layer for IEEE 802.11p

As aforementioned, the IEEE 802.11p standard is designed to operate in the DSRC band, unlike IEEE 802.11a which operates in 5GHz ISM band. The DSRC spectrum allocation is shown in Fig. 3-3. In USA the spectrum is structured into seven 10MHz channels. The central one is the Control Channel (CCH) and is restricted to safety-critical communications only. The first and the last channel are reserved for special uses. As an option two adjacent service channels may be used as one 20 MHz channel. The rest are services channels (SCH) available for both safety and non-safety usage and a 5MHz guard band. This distribution is shown in Fig. 3-2.

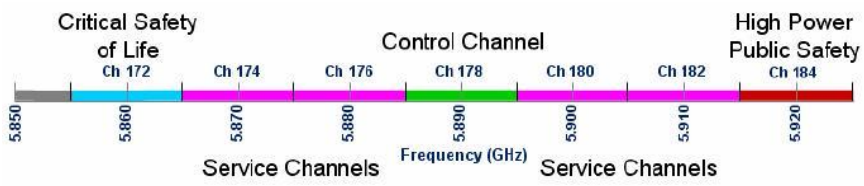


Fig. 3-2 US distribution of DSRC spectrum

The allocation of dedicated spectrum in Europe has been more difficult, due to the multiple parts involved. The Electronic Communications Committee (ECC) reserved in 2008 five channels of 10MHz. These channels are places in the frequency band between 5,875 and 5,925. This band is not exactly the same as in the US, however ECC recommends to use the spectrum between 5,855 - 5,875 for non-secure ITS applications.

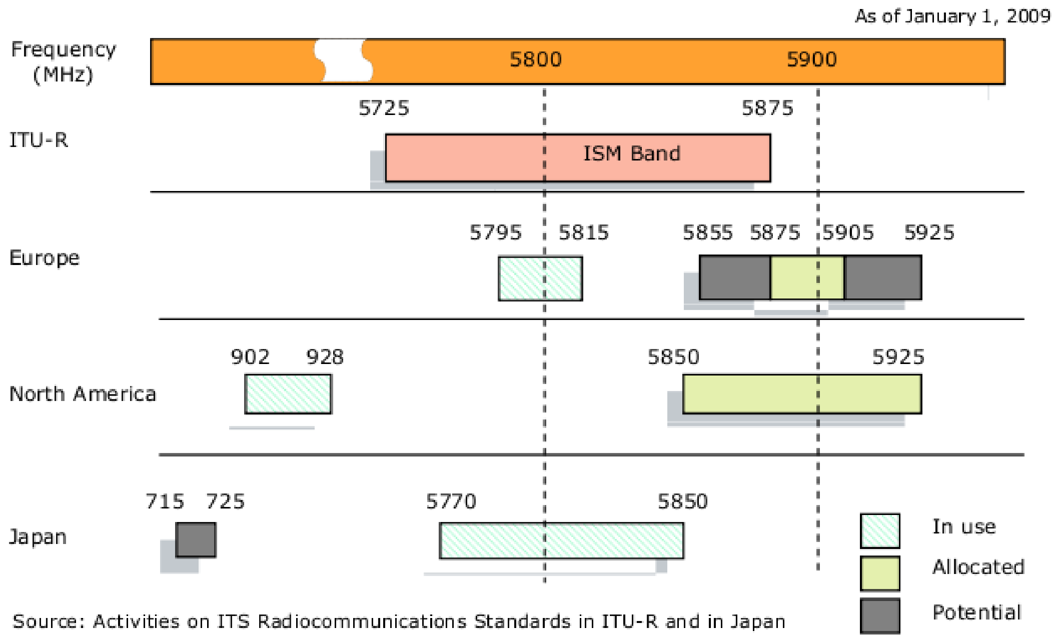


Fig. 3-3 DSRC spectrum allocation worldwide

The PHY layer of 802.11p takes almost the same signal processing and specifications from the 802.11a [85]. It is based on orthogonal frequency division multiplexing (OFDM), with some minor changes to fit the high-speed vehicular environment. The IEEE 802.11p together with the IEEE 1609.4 standard is designed for 10 MHz wide channels instead of 20 MHz as it is in the original 802.11a in order to increase the tolerance for multipath propagation. Due to this, the transfer rates will be halved in IEEE 802.11p compared to IEEE 802.11a, implying transfer rates of 3, 4.5, 6, 9, 12, 18, 24, and 27 Mbps. On the one hand this reduces the effects of Doppler spread by having a smaller frequency bandwidth; on the other hand the doubled guard interval reduces inter-symbol interference caused by multipath propagation.

Another two differences compared to the original IEEE 802.11 are presented below. Firstly, there is no difference between the nodes in the network, that is, all nodes are peers including the roadside units. There exists no access point functionality in IEEE 802.11p even though the vehicular network will contain roadside units at certain spots. And secondly, in order to support larger communication range in vehicular environments, four classes of maximum allowable Effective Isotropic Radiated Power (EIRP) up to 44.8 dBm (30W) are defined in IEEE 802.11p. The largest value is reserved for use by approaching emergency vehicles. A typical value for safety relevant messages is 33 dBm.

In summary, the main system parameters of the IEEE 802.11a and IEEE 802.11p are shown in Table. 3-2 to underline the differences between the two standards.

Parameter	***802.11a***	***802.11p***
Channel bandwidth	*20 Mhz*	*10 Mhz*
Chip duration	*50 ns*	*100 ns*
Number of fft points	*64*	*64*
Number of sub carriers	*52 + DC*	*52 + DC*
Number of data sub carriers	*52*	*52*
Number of pilot sub carriers	*4*	*4*
OFDM sympol period (80 chips)	*4 us*	*8 us*
Cyclic prefix (16 chips)	*0.8 us*	*1.6 us*
FFT Symbol period (64 chips)	*3.2 us*	*6.4 us*
Coding scheme	*½ industry convolutional*	*½*
Available data rate	*6-9-12-18-24-36-48-52 Mbps*	*3-4.5-6-9-12-18-24-27 Mbps*

Table 3-2 Comparison of the PHY layer implementation in IEEE 802.11a and IEEE 802.11p

3.2.2.2. MAC layer for IEEE 802.11p

The MAC method of the standard IEEE 802.11p is enhanced distributed channel access (EDCA) from the QoS amendment IEEE 802.11e as MAC method, which is an enhanced version of the basic distributed coordination function (DCF) found in IEEE 802.11 and based on CSMA/CA, meaning that the node or station starts by listening to the channel, and if it is free for a time period called an arbitration interframe space (AIFS), the sender can start transmitting directly. If the channel is busy or becomes occupied during the AIFS, the station must perform a backoff, that is, the node has to defer its access according to a randomized time period [10].

In 802.11p, QoS is obtained by putting the data traffic within each node into four different priority queues. These queues have different AIFS and backoff parameters, that is, the higher priority, the shorter AIFS. The backoff procedure in IEEE 802.11 works as follows:

- draw an integer from a uniform distribution [0, CW] , where CW refers to the current contention window
- multiply this integer with the slot time derived from the PHY layer in use, and set this as the backoff value
- decrease the backoff value only when the channel is free

- upon reaching a backoff value of 0, send immediately.

The MAC protocol of 802.11 is a stop-and-wait protocol and therefore the sender awaits an acknowledgment (ACK). If no ACK is received due to e.g., the transmitted packet never reaches the recipient, the packet being incorrect at reception, or the ACK being lost or corrupted, a backoff procedure is invoked before a retransmission is allowed. For every attempt to send a specific packet, the size of the CW will be doubled from its initial value (CWstart) until a maximum value (CWend) is reached. This is due to the fact that during high utilization periods, it is convenient to spread the nodes that want to send in time. After a successful transmission or when the packet had to be thrown away because the maximum number of channel access attempts was reached, the contention window will be set to its initial value again. In 802.11p different QoS classes are obtained by prioritizing the data traffic within each node. There are four different priority levels implying that each station maintains four queues. These queues have different AIFS and different backoff parameters, e.g., the higher priority, the shorter AIFS. In a broadcast situation, i.e., when packets destined for all nodes are transmitted, none of the receiving nodes will send ACKs in response. Therefore, a sender never knows if anyone has received the transmitted packet correctly, and it will perform at most one backoff (which occurs when a busy channel is sensed at the initial channel access attempt). Hence, at most one backoff decrement will take place for broadcasted packets [reference]. In Fig. 3-4 default parameter settings for the different queues in IEEE 802.11p are found together with the CW setting.

	Queue no. 1	**Queue no. 2**	**Queue no. 3**	**Queue no. 4**
Priority	Highest		→	Lowest
AIFS	34 μs	34 μs	43 μs	79 μs
CW_{start}	3	7	15	15
CW_{end}	511	1023	1023	1023

Fig. 3-4 Default parameter settings for IEEE 802.11p queues

4. ROUTING PROTOCOLS FOR AD HOC NETWORKS

Generally, routing protocols implemented for wired networks are not suitable for MANETs since they are based on periodic route updating mechanisms which increases overhead and cannot efficiently handle topology changes. Routing in MANETs and VANETs is complex since mobility causes frequent topology changes and requires more robust and flexible mechanism to search for routes and maintain them. When the network nodes move, the established paths may break and the routing protocols must dynamically search for other feasible routes. With a changing topology, even maintaining connectivity is very difficult. Therefore, routing protocols for MANETs and VANETs must deal with the following premises:

- **Distributed operation**, since is the basis of MANETs and VANETs
- **Signaling reduction**, allowing conserving battery capacity and enhancing network efficiency
- **Keeping the routes loop free**, in order to avoid packets flowing indefinitely on the network and network congestion
- **Reduced processing time**, aiming to save node's resources
- **Management of asymmetric links**, caused by different power levels among mobile nodes and other factors such as terrain conditions

4.1. Classification of routing protocols for ad hoc networks

Many protocols have been proposed for MANETs and VANETs. These protocols can be divided into three categories: proactive, reactive and hybrid. Proactive methods, also called *table-driven* methods, maintain routes to all nodes, including nodes to which no packets are sent. Such methods react to topology changes, even if no trafic is affected by the changes. Reactive methods, also called *on-demand* methods, are based on demand for data transmission. Routes between hosts are determinated only when they are explicity needed to forward packets. They can significantly reduce routing overhead when trafic is lightweight and the topology changes decrease dramatically, since they do not need to update route information periodically and do not need to find and mantain routes on which there is no trafic. Hybrid methods combine proactive and reactive methods to find efficient routes, without much control overhead.

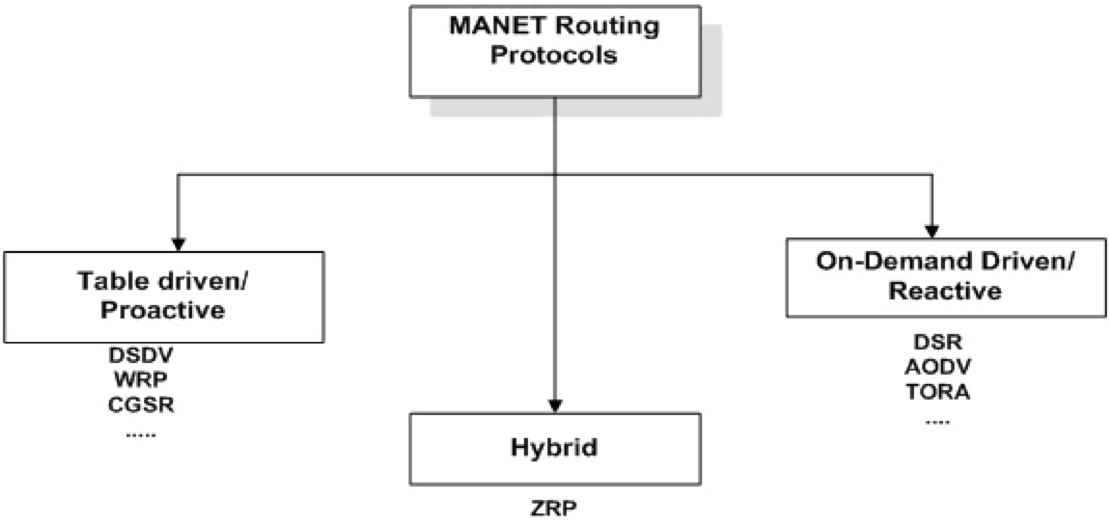


Fig. 4-1 Classification of routing protocols for MANETs and VANETs

4.1.1. Proactive Routing Protocols

As stated earlier, proactive routing protocols maintain routes to all destinations, regardless of whether or not these routes are needed. In order to maintain correct route information, a node must periodically send control messages. Therefore, proactive routing protocols may waste bandwidth since control messanges are sent out unnecessarily when there is no data traffic. The main advantage of this category of protocols is that hosts can quickly obtain route information and quickly establish a session. Several proactive routing protocols have been implemented depending on the kind of route information stored on node's tables as well as the used updating method. The most representative are DSDV (Destination-Sequenced Distance Vector) [8] and ADV (Adaptive Distance Vector) [12].

4.1.2. Reactive Routing Protocols

Reactive routing protocols can dramatically reduce routing overhead beacuse they do not need to search and maintain routes on which there is no data trafic at the expense of increasing end-to-end delay. This property is very appealing in the resource-limited environment. Depending on how the routing method is implemented, reactive routing protocols can be divided in *source routing* protocols and *hop-by-hop* or *point-to-point* protocols.

4.1.2.1. Source routing protocols

In source routing protocols every data packet carries the whole path information in its header. Before a source node sends data packets, it must know the total path to the destination, that is, all addresses of nodes which compose the path from source to destination. There is no need that intermediate nodes update its routing tables, since they only forward data packets according to the header information. However it entails scalability problems since as the number of hops increases, the path information every data packet must carry become major and it may waste bandwidth. Moreover, the path is established from the source node so that a bad adaptation to quickly topology changes will be performed.

The most representative source routing protocol is DSR (Dynamic Source Routing) [45].

4.1.2.2. Hop-by-hop routing protocols

Hop-by-hop routing protocols try to improve performance by keeping the routing information in each node. Every data packet does not include the whole path information any more. On the contrary they only include the address of the following node where data packet must be forwarded to get the destination as well as the destination address. Every intermediate node must look up its own routing table to forward the data packets to its destination, so that the route is calculated hop by hop. Hop-by-hop routing protocols save bandwidth and performs well in a large network since a data packet does not carry the whole path information. However, intermediate nodes must update their routing tables.

The most representative hop-by-hop routing protocol is AODV (Ad hoc On-demand Distance Vector) [65].

4.1.3. Hybrid Routing Protocols

Hybrid routing protocols combine the proactive and reactive routing approaches. They divide the network into routing zones, so that it will be used proactive routing schemes for intra-zones routing issues and reactive routing schemes for inter-zones routing issues.

The most representative hybric routing protocol is ZRP (Zone Routing Protocol) [73].

4.2. AODV

Ad hoc On-demand Distance Vector (AODV) routing protocol was motivated by the limited bandwidth thas is available in the media that are used for wireless communications. Unlike DSR routing protocol, AODV determines a route to a destination only when a node wants to send a packet to a destination. Routes are maintained as long as they are needed by the source [75].

4.2.1. Route discovery mechanisms

Route discovery process starts when a source node does not have routing information for a node to be communicated with. When a source node has to communicate with another destination node, it initiates path discovery by flooding a route request packet (RREQ) throughout the networks. The RREQ contains the fields shown in Fig.4-2.

source address	request ID	source sequence No.	destination address	destination sequence No.	hop count

Fig. 4-2 RREQ packet format

The *request ID* is incremented each time the source node sends a new RREQ, so the pair *(source address, request ID)* identifies a RREQ uniquely. On receiving a RREQ message each node checks the source address and the *request ID*. An intermediated node could receive several RREQ from different neighbors. However, if the node has already received a RREQ with the same pair of parameter the new RREQ packet will be discarded.

Source and destination sequence number prevent to use non-updated routes. Destiny sequence number identifies the most recent known route to reach this destiny. Source sequence number identifies the known route when RREP is transmitted from source. Thus, non-updates routes will never be considered.

As RREQ travels from node to node, it automatically sets up the reverse path from all these nodes back to the source. Each node that receives this packet records the address of the node from which it was received. This is called *Reverse Path Setup*. The nodes maintain this information for enough time for the RREQ to traverse the network and produce a reply to the sender and time depends on network size.

Every node which receives a RREQ checks if there is an entry for the desired destination in its routing table. If an entry is found and the destination sequence number is greater than that in the RREQ, this node unicasts a Route Reply Packet (RREP) to the neighbour node from which the RREQ was received. Otherwise, if the destination sequence number is lower than that in the RREQ or there is no entry for the desired destination in its routing table, the node rebroadcasts the RREQ to its neighbors. RREP message format is shown in Fig. 4-3.

source address	destination address	destination sequence No.	hop count	life-time

Fig. 4-3 RREP packet format

Once the RREP is generated, it travels back to the source, based on the reverse path. As the RREP travels back to source, each node along this path sets a forward pointer to the node from where it is receiving the RREP and records the latest destination sequence number to the request destination. This is called *Forward Path Setup*.

If an intermediate node receives another RREP after propagating the first RREP towards source, it checks for destination sequence number of new RREP. The intermediate node updates routing information and propagates new RREP only if the destination sequence number is greater or the new sequence number is same and hop count is lower. Otherwise, it just skips the new RREP. This ensures that algorithm is lopp-free and only the most effective route is used.

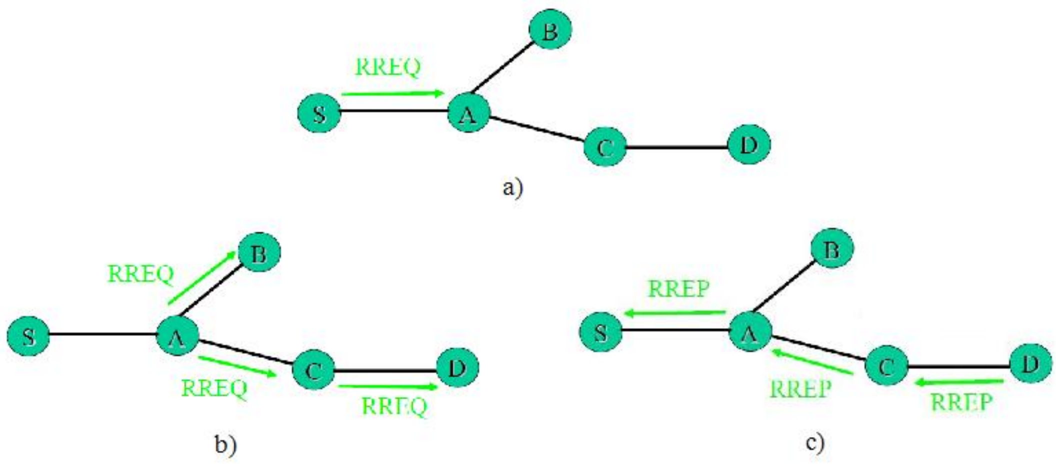


Fig. 4-4 Route discovering mechanism, a) Source node initiates path discovering sendig RREQ message *b) Node A broadcasts RREQ message to its neighbor nodes c) Destiny node unicasts a RREP message to node C in order to reach source node*

4.2.2. Route maintenance mechanisms

Each node can get to know its neighborhood by using local broadcasts, so-called *hello messages*. Nodes neighbors are all the nodes that it can directly communicate with. Although AODV is a reactive protocol it uses these periodic *hello messages* to inform the neighbors that the link is still alive. The Hello messages will never be forwarded because they are broadcasted with TTL = 1. When a node receives a *hello message* it refreshes the corresponding lifetime of the neighbor information in the routing table.

If a node has received no messages from some outgoing node for an extended period of time, then that node is presumed to be no longer reachable. Whenever a node determines one of its next hops to be unreachable, it removes all affected route entries, and generates a Route Error message (RERR). This RERR message contains a list of all destinations that have become unreachable as a result of the broken link. The node sends the RERR to each of its precursors. These precursors update their routing tables, and in turn forward the RERR to their precursors, and so on, until the source node is reached.

When the source node receives the link failure notification message, it will re-initiate a route discovery for the destination if a route is still needed. A new destination sequence number is used to prevent routing loops formed by the entangling of obsolet and new established paths.

To prevent RERR message loops, a node only forwards a RERR message if at least one route has been removed.

4.2.3. Comparative: AODV vs DSR

The following section is dedicated to illustrate existing similarities and differences between the two most representative reactive routing protocols, AODV and DSR, as well as their advantages and disadvantages. Table 4-1 shows a complete comparative between the features of both protocols.

<table>
<tr><th>Features</th><th>DSR</th><th>AODV</th></tr>
<tr><td>Mechanism</td><td>Reactive source routing</td><td>Reactive hop-by-hop routing</td></tr>
<tr><td rowspan="3">Routes</td><td>Multiple routes for each destiny</td><td>Single route for each destiny</td></tr>
<tr><td>Each packet includes all path information</td><td>Each packet only includes source and destiny address</td></tr>
<tr><td>Only source node must update its route table and stores all available routes</td><td>All nodes must update their route tables and stores only the newest route</td></tr>
<tr><td rowspan="3">Discovering procedure</td><td colspan="2">Discovering mechanism built on request messages</td></tr>
<tr><td>Generates more loads of data packets since all existing routes are stored</td><td>Generates less loads of data packets since there is only one stored route for each destiny</td></tr>
<tr><td colspan="2">Discovery mechanism only starts when a source node has data to transmit</td></tr>
<tr><td rowspan="2">Route management</td><td>Link failures are reported to source node</td><td>Link failures are reported to the precursor node</td></tr>
<tr><td>Source node is concerned about how new/old routes are</td><td>A sequence number is required to distinguish how new/old routes are</td></tr>
</table>

	Only source node takes part in routing path decision	*All nodes take part in routing path decision*
	Obsolet routes are still available	*Only the newest route is available*
Performance specifications	*Better end-to-end delay and latency in lightweight traffic and low mobility scenarios*	*Better end-to-end delay and latency in heavyweight traffic and high mobility scenarios*
	Higher overhead but low signalign traffic	*Lower overhead but higher signaling traffic*
	Simetric links are not required	*Simetric links are required*

Table 4-1 Comparative AODV vs DSR

5. PROPAGATION MODELS FOR VEHICULAR ENVIRONMENTS

Radio propagation is usually attributed to three mechanisms: reflection, diffraction and scattering. If a line of sight (LOS) exists between the transmitter and receiver (T-R), then reflection will dominate. On the other hand, in an obstructed scenario, diffraction and scattering will be the main cause of interference. Due to these characteristics, the channel varies rapidly with time and the user's location, phenomena known as fading.

There are two types of fading: large-scale and small-scale fading. Large-scale fading correlates to shadowing while small-scale fading correlates to multi-path propagation delays. In the following, an overview of both fading types is presented.

5.1. Large-Scale Fading Models: Two-Ray ground model

Large-scale propagation models describe attenuation of transmitted signal over large T-R distances. A well known model is the Free Space model [56], which describes power transfer between transmitter and receiver over free space. Free Space model only takes into account LOS waves from the transmitter.

A more accurate model called Two-Ray ground model considers waves reflected from ground in addition to LOS waves. The Two-Ray ground model provides a more accurate model compared to Free Space Model since waves naturally reflect off surrounding objects. Given the distance between the transmitter and the receiver, d, the expression for the Two-Ray ground model is as follows:

$$P_r(d) = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4} \tag{5.1}$$

where P_t is the transmitted power, G_t and G_r (both having dimensionless quantities) are the antenna gains of the transmitter and receiver respectively and h_t and h_r are the heights (in meter) of the transmitter and receiver from the ground respectively.

5.2. Small-Scale Fading Models

The previous models do no describe the rapid fluctuations of the received signal due to multipath fading. In obstructed environment signals fade faster due to reflection, refraction and scattering resulting in the transmitted signal taking multiple paths to the receiver. Each separate path is attenuated depending on the passage taken. The combination of directed and out-of-phase reflected waves at the receiver yields attenuated signals, phenomena known as the aforementioned multipath fading. The relative speed between the transmitter node and the received node or the surrounding objects causes fading too. The main multipath fading effects are:

- Rapid changes in signal strength over a small area or time interval
- Random frequency modulation due to varying Doppler shifts on different multipath signals
- Time dispersion or echoes caused by multipath propagation delays

The most popular small-scale fading models, Rayleigh model and Rician model, are descripted below.

5.2.1. Rayleigh model

A common model used to describe small-scale fading is the Rayleigh distribution. The Rayleigh probability density function (PDF) is given by:

$$P_r(r) = \frac{r}{\sigma^2} \exp\left[-\frac{r^2}{2\sigma^2}\right], \qquad r \geq 0 \tag{5.2}$$

where σ is the Rayleigh parameter or the root mean square (RMS) of the received voltage signal before envelope detection [70].

The main limitation of the Rayleigh model is that it assumes that all transmitted signal arriving at the receiver are attenuated equally.

As mentioned, the Rayleigh model describes the reception of N multi-path waves at the receiver having equal power. Given large N, it can be shown that through the central limit theorem that the in-phase and quadrature components of the signal tend to be Gaussian of zero-mean. Thus, the received signal $r(t)$ can be modeled using the following equation:

$$r(t) = \sqrt{x(t)^2 + y(t)^2} \tag{5.3}$$

where $x(t)$ and $y(t)$ are Gaussian random variable.

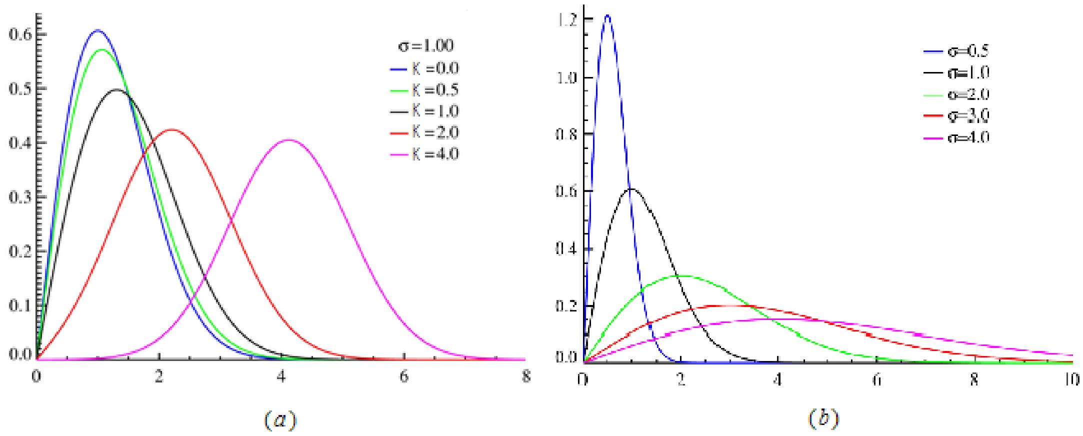


Fig. 5-1 Small-scale fading distribution: (a) Rician (b) Rayleigh

5.2.2. Rician model

The Rician model is similar to Rayleigh model, except that in Rician model one of the path, typically a LOS signal, is much stronger than the others. The Rice probability density function (PDF) of the received signal envelope, $R(t)$ is given by

$$P_r(r) = \frac{2(K+1)r}{\Omega}\exp\left(-K - \frac{(K+1)r^2}{\Omega}\right) I_0\left(2\sqrt{\frac{K(K+1)}{\Omega}}r\right) \tag{5.4}$$
$$r \geq 0, \quad K \geq 0; \quad \Omega \geq 0$$

where $I_n(\cdot)$ is the nth-order modified Bessel function of the first kind [1]. The parameter *K* is the ratio of the power received via the LOS path to the power contribution of the non-LOS paths, and is a measure of fading whose estimate is importany in link budget calculations (notice that Rician model derives in Rayleigh model considering K=0, that is, the dominant LOS compoment fades away). Ω is the power in the direct path, and acts as a scaling factor to the distribution.

Both Rayleigh and Rician propagation models are commonly used to describe MANET and VANET environments [78]. Fading with the Rayleigh distribution is for highly mobile conditions with NLOS between nodes, while latter is suitable where there is a direct LOS path between transmitter and receiver.

6. SIMULATION TOOLS

The following tools take part in the VANET simulation described in the current document: VanetMobiSim, NS2 and AWK. All three are open source applications and are descripted in more detail below.

6.1. VanetMobiSim

6.1.1. Introduction

The Vehicular Ad Hoc Network Mobility Simulator (VanetMobiSim) is a set of extensions to CanuMobiSim, a framework for user mobility modeling used by the CANU (Communications in Ad Hoc Networks for Ubiquitous Computing) Research group [15], University of Stuttgart. The framework includes a number of mobility models, as well as parsers for geographic data sources in various formats, and a visualization module. The framework is based on the concept of pluggable modules so that it is easily extensible.

The set of extensions provided by VanetMobiSim consists mainly on the two following:

A vehicular spatial model, composed of spatial elements (such as traffic lights or multi-lane roads), their attributes and the relationships linking these spatial elements in order to describe vehicular areas. The spatial model is created from topological data obtained in four different ways:

- *User-defined* – The user defines a set of vertices and edges composing the backbone of the vehicular spatial model.
- *Random* – The backbone is randomly generated using the Voronoi tesseletions.

- *Geographic Data Files (GDF)* – Backbone data is obtained from GDF files.
- *TIGER/Line Files* – Similar to previous one, but based on the TIGER/line files from the US Census Bureau [18].

A set of vehicular-oriented mobility models, whose main compoments are the support of a microscopic level mobility models:

- *Intelligent Driving Model with Intersection Management (IDM_IM)*, describing perfectly car-to-car and intersection managements.
- *Intelligent Driving Model with Lane Changing* (IDM_LC), an overtaking model is also included, which interacts with IDM_IM to manage lanes changes and vehicle accelerations and decelerations.

VanetMobiSim offers so many possibilities and features to create realistic scenarios. Besides that, simulation scenarios for VanetMobiSim are defined in XML format using tags, making scenario configuration easier and in a more handy way. Definitely, VanetMobiSim is quite more appropiate in order to generate scenarios for VANETs than other MANETs mobility pattern generators such as CityMob [21] and Bonnmotion [11].

6.1.2. How to install VanetMobiSim-1.1

The first step is downloading the source code of VanetMobiSim-1.1 from [80] and expand it in a base directory of your choice

```
# unzip ~/Descargas/VanetMobiSim-1.1.zip -d ~/VanetMobiSim
```

The previous command unzips the source code of *VanetMobiSim-1.1* downloaded to the *Downloads Folder* to a directory called *VanetMobiSim* in our personal directory. The following subdirectories and files will be created after extraction

```
/ jar
build.xml
 VanetMobiSim-src.jar
VanetMobiSim-samples.jar
mypackages.lst
READ_ME
```

In the next step you must download the source code of CanuMobiSim v1.3.4 from [15] and expand it in the same directory

```
# unzip ~/Descargas/CanuMobiSim_1_3_4_src.zip -d ~/VanetMobiSim
```

You should get a subdirectoy named */src* so that at this time, your current directory should contains

```
/ jar
/src
build.xml
 VanetMobiSim-src.jar
VanetMobiSim-samples.jar
mypackages.lst
READ_ME
```

At this point, VanetMobiSim requires the presence of Sun JVM, a virtual machine capable of executing Java bytecode, as well as Apache Ant, a tool whose mission is to build Java applications. The following command will install the open-ource Java Development Kid (JDK) v.6

```
# sudo apt-get install openjdk-6-source
```

For installing Apache Ant v1.7 the following command must be type

```
# sudo apt-get install ant1.7
```

The next step requires going to the VanetMobiSim directory and launch ant

```
# cd ~/VanetMobiSim

# ant patch
```

Ant will patch the */src* directory with *VanetMobiSim* source files. The message appeared in the terminal after patching is shown in Fig. 6-1.

Fig. 6-1 Terminal message after executing ant patch

Eventyally you must type the following command in order to build the simulator and create the javadocs

```
#ant all
```

The message appeared in the terminal after building is shown in Fig. 6-2. The binary *.jar* file of VanetMobiSim will be place in the */jar* subdirectory. Now *VanetMobiSim is available to use*.

```
Buildfile: build.xml

init:
    [mkdir] Created dir: /home/rcal2720/VanetMobiSim/classes

compile:
    [javac] Compiling 127 source files to /home/rcal2720/VanetMobiSim/classes
    [javac] Note: Some input files use unchecked or unsafe operations.
    [javac] Note: Recompile with -Xlint:unchecked for details.

jars:
      [jar] Building jar: /home/rcal2720/VanetMobiSim/jar/VanetMobiSim.jar

javadocs:
    [mkdir] Created dir: /home/rcal2720/VanetMobiSim/doc/api
  [javadoc] /home/rcal2720/VanetMobiSim/src/CanuMobiSim contains source files in
 the default package, you must specify them as source files not packages.
  [javadoc] /home/rcal2720/VanetMobiSim/src/CanuMobiSimAdd contains source files
 in the default package, you must specify them as source files not packages.
  [javadoc] Generating Javadoc
  [javadoc] Javadoc execution
```

Fig. 6-2 Terminal message after executing ant all

6.2. NS2

6.2.1. Introduction

The Network Simulator 2 (also popularly called NS2) is a discrete event network simulator targeted at networking research. NS provides a packet level simulation over a lot of protocols, supporting several transport protocols, several forms of multicast, wired networking, several ad-hoc routing protocols and propagation models, data broadcasting, satellite and so on [63]. Also, NS2 has the possibility of using mobile nodes. The mobility of these nodes may be specified either directly in the simulation file or by using a mobility trace file. In our case, the trace file is generated by VanetMobiSim. Nearly all mobility pattern simulator (including the aforementioned VanetMobiSim, CityMob and Bonnmotion) are implemented in order to produce mobility trace files absolutely compatible with NS2 contrary to others network simulator such as NCTUns [59], so that a whole simulation running both VanetMobiSim and NS2 together can be carried out with total ease. Hence it is heavily used in ad hoc networking research and has become popular in research due to its open source model and online documentation. The last released version of this network simulator is NS2.34 (from Jun 17 2009).

NS began as a variant of the REAL network simulator in 1989 and has evolved substancially over the past few years. In 1995 NS development was supported by DARPA through the VINT [81], a collaborative project at LBNL (*Lawrence Berkeley National Laboratory)*, Xerox PARC (*Xerox Palo Alto Research Center)*, UCB (*University of California*, Berkeley), and USC/ISINS (*University of Southern California's Information Sciences Institute)*.

NS was built in C++ and provides a simulation interface through OTcl, an object-oriented dialect of Tcl (Tool Command Language). The user describes a network topology by writing

OTcl scripts, and then the main NS program simulates that topology with specified parameters. Moreover, NS2 is easily extensible since the simulation kernel source code is available, which allows to implement new routing protocols, propagation models and so on, and use them in our simulations.

6.2.2. How to install NS2.34

First of all, you must download the NS2.34 source code from [62] and extract it in a base directory on your choice, e.g. your personal directory

```
# tar -xzf ns-allinone-2.34.tar.gz -C ~/
```

The directory */ns-allinone-2.34* will be created after extraction

Before start installation the required build-essential, libX11-dev and xorg-dev packets must be installed, typing

```
# sudo apt-get install build-essential

# sudo apt-get install libX11-dev

# sudo apt-get install xorg-dev
```

Now the installation can be started. Go to the directory /ns-allinone-2.34 and start installation

```
# cd ~/ns-allinone-2.34

# ./install
```

After a few minutes and if installation have succeed a message will be appear on your terminal as shown in Fig. 6-3. However, you probably may encounter the error shown in Fig. 6-4

```
--------------------------------------------------------------------------------
--
Please put /home/rcal2720/ns-allinone-2.34/bin:/home/rcal2720/ns-allinone-2.34/t
cl8.4.18/unix:/home/rcal2720/ns-allinone-2.34/tk8.4.18/unix
into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.

IMPORTANT NOTICES:

(1) You MUST put /home/rcal2720/ns-allinone-2.34/otcl-1.13, /home/rcal2720/ns-al
linone-2.34/lib,
    into your LD_LIBRARY_PATH environment variable.
    If it complains about X libraries, add path to your X libraries
    into LD_LIBRARY_PATH.
    If you are using csh, you can set it like:
                setenv LD_LIBRARY_PATH <paths>
    If you are using sh, you can set it like:
                export LD_LIBRARY_PATH=<paths>

(2) You MUST put /home/rcal2720/ns-allinone-2.34/tcl8.4.18/library into your TCL
_LIBRARY environmental
    variable. Otherwise ns/nam will complain during startup.

After these steps, you can now run the ns validation suite with
cd ns-2.34; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mailing list ar
chive
for related posts.
```

Fig. 6-3 Terminal message after installation succeed

```
gcc -c -g -O2 -DNDEBUG -DUSE_SHM -fpic -I. -I/home/rcal2720/ns-allinone-2.34/include -I/ho
34/include -I/include  otcl.c
ld -shared -o libotcl.so otcl.o
otcl.o: In function `OTclDispatch':
/home/rcal2720/ns-allinone-2.34/otcl-1.13/otcl.c:495: undefined reference to `__stack_chk_
otcl.o: In function `Otcl_Init':
/home/rcal2720/ns-allinone-2.34/otcl-1.13/otcl.c:2284: undefined reference to `__stack_chk
ld: libotcl.so: hidden symbol `__stack_chk_fail_local' isn't defined
ld: final link failed: Nonrepresentable section on output
make: *** [libotcl.so] Error 1
otcl-1.13 make failed! Exiting ...
See http://www.isi.edu/nsnam/ns/ns-problems.html for problems
rcal2720@rcal2720-laptop:~/ns-allinone-2.34$
```

Fig. 6-4 OTCL error message while installing

This error occurs because the linker being used is *ld-shared* instead of *gcc-shared*. You must edit one line in */ns-allinone-2.34/otcl-1.13/configure* as follows

```
--- configure.orig     2009-11-02 12:14:52.556167945 -0800
+++ configure   2009-11-02 12:17:28.966706099 -0800
@@ -6301,7 +6301,7 @@
    ;;
 Linux*)
       SHLIB_CFLAGS="-fpic"
-      SHLIB_LD="ld -shared"
+      SHLIB_LD="gcc -shared"
       SHLIB_SUFFIX=".so"
```

DL_LIBS="-ldl"

SHLD_FLAGS=""

Furthermore, if the required packet installation presented in step 2 is skipped the following error messages will appear on your console

```
g++   -o tcl2c++ tcl2c++.o
make: g++: No se encontró el programa
make: *** [tcl2c++] Error 127
tclcl-1.19 make failed! Exiting ...
See http://www.isi.edu/nsnam/ns/ns-problems.html for problems
rcal2720@rcal2720-laptop:~/ns-allinone-2.34$
```

Fig. 6-5 Error message while installing without build essential packet

```
gcc -c -O2  -pipe  -Wall -Wno-implicit-int -fno-strict-aliasing -fPIC -I/home/rcal2720/ns-allinone-2.34/tk8.4.18
/unix -I/home/rcal2720/ns-allinone-2.34/tk8.4.18/unix/../generic -I/home/rcal2720/ns-allinone-2.34/tk8.4.18/unix
/../bitmaps -I/home/rcal2720/ns-allinone-2.34/tcl8.4.18/generic  -DHAVE_LIMITS_H=1 -DHAVE_UNISTD_H=1 -DSTATIC_BU
ILD=1 -DPEEK_XCLOSEIM=1 -D_LARGEFILE64_SOURCE=1 -DTCL_WIDE_INT_TYPE=long\ long -DHAVE_STRUCT_STAT64=1 -DHAVE_OPE
N64=1 -DHAVE_LSEEK64=1 -DHAVE_TYPE_OFF64_T=1 -DHAVE_SYS_TIME_H=1 -DTIME_WITH_SYS_TIME=1 -DSTDC_HEADERS=1 -DHAVE_
PW_GECOS=1      -DTCL_NO_DEPRECATED    /home/rcal2720/ns-allinone-2.34/tk8.4.18/unix/../generic/tk3d.c
In file included from /home/rcal2720/ns-allinone-2.34/tk8.4.18/unix/../generic/tkInt.h:21,
                 from /home/rcal2720/ns-allinone-2.34/tk8.4.18/unix/../generic/tk3d.h:18,
                 from /home/rcal2720/ns-allinone-2.34/tk8.4.18/unix/../generic/tk3d.c:16:
/home/rcal2720/ns-allinone-2.34/tk8.4.18/unix/../generic/tk.h:81: fatal error: X11/Xlib.h: No existe el archivo
o directorio
compilation terminated.
make: *** [tk3d.o] Error 1
tk8.4.18 make failed! Exiting ...
For problems with Tcl/Tk see http://www.scriptics.com
rcal2720@rcal2720-laptop:~/ns-allinone-2.34$
```

Fig. 6-6 Error message while installing without libX11-dev packet

```
checking for http.tcl... ../lib/tcl8.4/http1.0
checking Tcl http.tcl library... yes
checking for tclsh8.4.18... no
checking for tclsh8.4... ../bin/tclsh8.4
checking for tk.h... -I../include
checking for libtk8.4... -L../lib -ltk8.4
checking for tk.tcl... ../lib/tk8.4
checking for X11 header files
can't find X includes
otcl-1.13 configuration failed! Exiting ...
Please check http://www.isi.edu/nsnam/ns/ns-problems.html
for common problems and bug fixes.
rcal2720@rcal2720-laptop:~/ns-allinone-2.34$ ^C
rcal2720@rcal2720-laptop:~/ns-allinone-2.34$
```

Fig. 6-7 Error message while installing without Xorg-dev packet

Now you can restart installation

```
# ./install
```

Once the installation process has finished, you must edit your path. Type the following command

```
# gedit ~/.bashrc
```

and add the following lines at the end of the file (remember replace */home/rcal2720* by the folder where you have extracted the NS2 files)

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/rcal2720/ns-allinone-2.34/otcl-1.13
NS2_LIB=/home/rcal2720/ns-allinone-2.34/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
GCC=/usr/bin
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_LIB:$
GCC

# TCL_LIBRARY
TCL_LIB=/home/rcal2720/ns-allinone-2.34/tcl8.4.18/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

# PATH
XGRAPH=/home/rcal2720/ns-allinone-2.34/bin:/home/route/ns-allinone-
2.34/tcl8.4.18/unix:/home/rcal2720/ns-allinone-2.34/tk8.4.18/unix
NS=/home/rcal2720/ns-allinone-2.34/NS2.34/
NAM=/home/rcal2720/ns-allinone-2.34/nam-1.14/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

Finally type the following command

```
# source ~/.bashrc
```

At this point the last step is to validate ns2

```
# cd ~/ns-allinone-2.34/NS2.34

# ./validate
```

Now NS2 is available to use.

6.3. AWK

6.3.1. Introduction

The AWK utility is a data extraction and reporting tool that uses a data-driven scripting language for the purpose of producing formatted reports. In other words, AWK is an excellent filter and files of text processor. It was created at Bell Labs in 1970s, and its name is derived from the family names of its authors – Alfred Aho, Peter Weinberger and Brian Kernighan.

A file is treated as a sequence of records, and by default each line is a record. Each line is broken up into a sequence of fields, so we can think of the first word in a line as the first field, the second word as the second field, and so on. AWK reads the input a line at a time. Then a line is scanned for each pattern in the program, and for each pattern that matches, the associated action is executed.

AWK is easier to use than most conventional programming languages. It can be considered to be a pseudo-C interpretor, as it understands the same arithmetic operators as C. AWK also has string manipulation functions, so it can search for particular strings and modify the output.

6.3.2. How to install AWK

For installing AWK, you must type the following command

```
# sudo apt-get install gawk
```

And now awk is available to use

6.4. Simulation process

In Fig. 6-8 the VANET simulation scheme is shown. After definating a mobility scenario in a XML file, launching the VanetMobiSim framework is necessary in order to produce a node mobility trace file in NS2 format (node identifier, time, position, speed). This file must be incorporate to the communications definition file, implemented in Tcl language. After running NS2 a trace file which logs all routing events during simulation is generated. Eventually, AWK tool is needed to filter that events trace file extracting all significant data, allowing an evaluation of the scenario. A step-by-step simulation review is presented in *Chapter 10*.

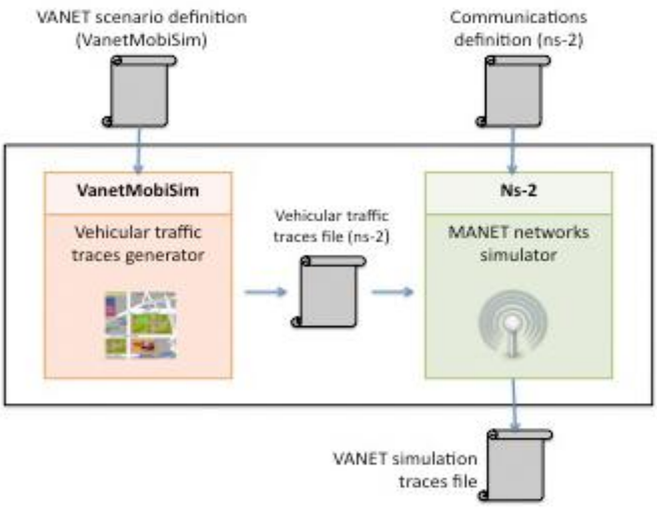


Fig. 6-8 VANET simulator based on coupling VanetMobiSim and NS2 [60]

7. DESCRIPTION OF THE SIMULATIONS

This section aims to present the scenarios and network features used to implement the simulations. As many empirical works, an initial configuration was firstly proposed and gradually was fitted using trial and error method in order to adjust to VANET environments as accurate as possible. In Annex 1 a complete simulation is presented, emphasizing on how to configure the scenarios and run the simulators.

7.1. Scenarios

During simulation process four scenarios have been implemented in order to use all the possibilities VanetMobiSim [80] offers. These scenarios are:

- Manhattan grid scenario
- Urban scenario with random street configuration
- Real urban scenario extracted from TIGER/Line [18] maps
- Real highway scenario extracted from TIGER/Line maps

A picture of each scenario is shown in *Fig.7- 1*.

In the urban scenarios (i, ii and iii) a 500x500 m^2 dimension has been selected, including 10 traffic lights with step of 20 seconds. In addition, the scenarios have been configured with single-lane as well as multi-lane lanes and differentiate two traffic flows. Regarding nodes, several speeds have been selected as well as level of congestion. In addition, node's motion is enhanced with Intelligent Driving Model (IDM), which incorporates intersection management and lane changing.

A highway scenario has also been implemented besides the urban scenarios previous mentioned. The purpose is not exhaustively analyzing this scenario but evaluate briefly its performance. The configuration of the scenario is quite similar to the urban scenario except that now dimension is 1000x1000 m² and no traffic lights have been incorporated. In addition, highway has been implemented with four lanes for each traffic flow. *Table.7-1* summarizes the features of each scenario.

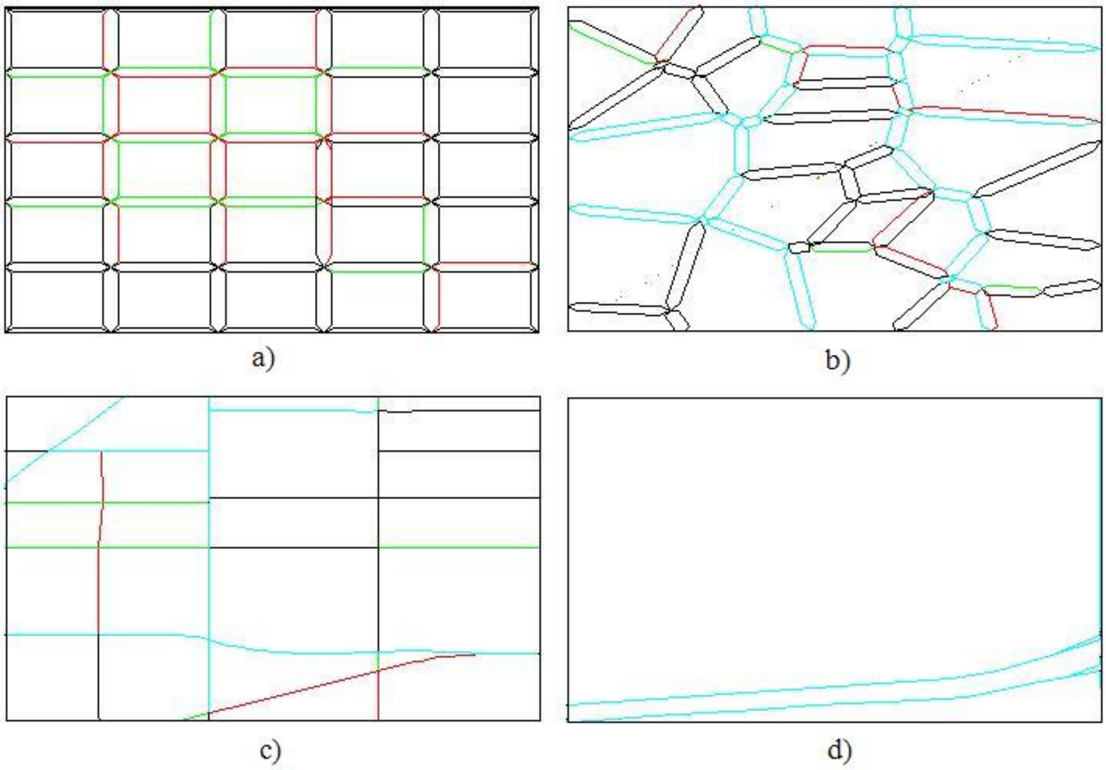


Fig. 7-1 Scenarios a) Manhattan grid b) Urban random distribution c) Urban TIGER/Line d) Highway TIGER/Line

For Urban TIGER/Line scenario, a downtown area from Washington D.C. is selected. On the other hand, the Highway TIGER/Line scenario corresponds to Highway nº 95 over the Woodrow Wilson Memorial Bridge in Maryland State.

	Manhattan	*Urban Random*	*Urban Tiger/Line*	*Highway Tiger/Line*
Dimensions (m^2)	*500x500*	*500x500*	*500x500*	*1000x1000*
Nodes	*30 / 60 / 90 / 120*	*30 / 60 / 90 / 120*	*30 / 60 / 90 / 120*	*30 / 60 / 90 / 120*
Traffic lights	*10 with step=20ms*	*10 with step=20ms*	*10 with step=20ms*	-

Lane Configuration	Differentiate two flows 5 streets with 2 lanes (the rest only with 1 lane)	Differentiate two flows 5 streets with 2 lanes (the rest only with 1 lane)	Differentiate two flows 5 streets with 2 lanes (the rest only with 1 lane)	Differentiate two flows Highway with 4 lanes for each flow
Speed of nodes (km/h)	5 / 35 / 65	5 / 35 / 65	5 / 35 / 65	60-120
Mobility Pattern	Activity-based travel Demand Model [64]	Random Waypoint Mobility with obstacle Avoidance [44]	Random Waypoint Mobility with obstacle Avoidance [44]	Random Waypoint Mobility with obstacle Avoidance [44]
Driving Model	Intelligent Driving Model (IDM) with intersection management and lane changing	Intelligent Driving Model (IDM) with intersection management and lane changing	Intelligent Driving Model (IDM) with intersection management and lane changing	Intelligent Driving Model (IDM) with intersection management and lane changing

Table 7-1 Scenario features overview

7.2. Parameters of the simulations

In this section the common used values as well as network configuration of the simulations in NS2 are presented.

- Simulation time is set to 2500 seconds. Nodes motion starts at t=0s and transmission starts at t=100s.
- Density of nodes has been set to 30, 60, 90 and 120. In all simulations there are only one source node and one destination node.
- Source Traffic is configured as a CBR/UDP agent with a rate of 20kbps, packet length of 512bytes.
- The queue selected is PriQueue (Queue/DropTail/PriQueue) which gives priority to routing packets. The queue size is set to 50 packets.
- Omni-directional antennas are selected with height of 1,5m.
- IEEE 802.11b and IEEE 802.11p have been selected as MAC protocols.
- Four propagation models have been selected. Two-Ray Ground model, LOS model and Ricean and Rayleigh models. In the latest version of NS2 (NS2.34) only the Two-Ray Ground is implemented. However many propagation models are available as ns2 extensions. LOS model extension is downloadable from [49] and Ricean and Rayleigh model extension is downloadable from [69].
- AODV is the routing protocol used in the simulations.
- Transmission range is set to 100, 125, 150, 175 and 200 meters. The sensing range is set to $0,78 \cdot \text{transmission range}$.
- Finally, each simulation has been carried out with 5 different seeds to obtain more accurate results.

The selected simulation values for each scenario are shown in *Table.7-3*.

	Manhattan	**Urban Random**	**Urban Tiger/Line**	**Highway Tiger/Line**
Simulation time	2500 sec	2500 sec	2500 sec	2500 sec
Nodes	30 / 60 / 90 / 120	30 / 60 / 90 / 120	30 / 60 / 90 / 120	30 / 60 / 90 / 120
Traffic Agent	CBR / UDP	CBR / UDP	CBR / UDP	CBR / UDP
Packet Lenght	512 bytes	512 bytes	512 bytes	512 bytes
Data Rate	20 kbps	20 kbps	20 kbps	20 kbps
Queue	PriQueue with size of 50 packets	PriQueue with size of 50 packets	PriQueue with size of 50 packets	PriQueue with size of 50 packets
Antenna	Omni-directional with height of 1,5m	Omni-directional with height of 1,5m	Omni-directional with height of 1,5m	Omni-directional with height of 1,5m
MAC Protocol	IEEE 802.11b / IEEE 802.11p	IEEE 802.11b	IEEE 802.11b	IEEE 802.11b
Propagation Model	TwoRayGround Line Of Sight (LOS) Ricean Rayleigh	TwoRayGround	TwoRayGround	TwoRayGround
Transmission Range	100 / 125 / 150 / 175 / 200	125	125	125
Routing Protocol	AODV	AODV	AODV	AODV
Number of seeds	5	5	5	5

Table 7-2 Simulation features overview

7.3. Evaluated metrics

In order to analyze the performance of vehicular ad hoc networks, extracting some information from the simulations is required. As stated in *Chapter 6*, an AWK filter must be implemented so that some metrics from traces files generated during simulation process can be studied. Our AWK filter has been designed to extract the following metrics:

7.3.1. Received Data Packet Ratio (RDPR)

It considers the percentage of data packets received by destination node relative to data packets sent by source node. In order to calculate this metric, it must defined two counter, C_S

and C_R . The first one should be increased for every packet sent by source node. The latter should be increased for every packet received by destination node. Eventually, the data packet received ratio is defined by

$$RDPR = \frac{\#received\_packets}{\#send\_packets} \tag{7.1}$$

7.3.2. Average End-to-end delay (E2E)

It considers the time which a packet needs to go through the network from source node to destination node. Once a packet with an *ID i* is sent by source node, the current time is stored in the position *i* of an array $(sent\_array[i])$. If the packet with *ID i* is received by destination node, the current time is stored in a second array $(received\_array[i])$. The average end-to-end delay is calculated by

$$E2E = \frac{\sum_{j \in received\ packets} (received\_array[j] - sent\_array[j])}{\#received\_packets} \tag{7.2}$$

7.3.3. Average number of hops (HOPS)

It considers the number of nodes that a packet must hop in order to reach the destination node. Every time a *RREP* message is received by source node, it must read its TTL value so that the number of hops used for this route can be deduced. Then a counter stores the number of packets received by destination node until the route is unavailable. This mechanism must be repeated for each valid route. The average number of hops is defined as the following expression

$$HOPS = \frac{\sum_{j}^{N_{ROUTES}} (hops)_j (\#received\_packets)_j}{\#received\_packets} \tag{7.3}$$

7.3.4. Data throughput

It considers the ratio between the total bps that source node is able to inject to the network and reach destination node. Our simulation time is 2400 seconds so it must be divided into 4 periods of 600 seconds. It have calculated data throughput as a time function, considering intervals of 600 seconds. Data throughput is defined as stated below

$$THROUGHPUT[t] = \frac{\#received\_bits[t]}{t} \tag{7.4}$$

8. SIMULATION RESULTS

This chapter has the aim of analyzing the results obtained from a large number of simulations run with NS2. They are structured as follows:

Firstly the **Manhattan scenario** is evaluated studying the global metrics and showing how the system performance is affected by configured parameters, e.g. node's density or transmission range. For this evaluation, the selected propagation model is **Two Ray Ground** and a comparison of MAC protocols is presented. The performance differences between **IEEE 802.11b and IEEE 802.11p** protocols described in *Chapter 3* are studied.

Afterwards, different **propagation models** are studied. The objective is to evaluate how the communication features of our Manhattan scenario are affected by each of these propagation models and comparing the results to what was described in *Chapter 5*.

Eventually, the other three scenarios are considered: urban with random street distribution scenario, real urban scenario extracted from **TIGER/Line database** and highway scenario also extracted from TIGER/Line database. A comparision between these scenarios is carried out, showing how their performances vary depending on the node's configuration.

Over 900 simulations have been run in order to carry out the following evaluation, without considering another few hundred simulations used to test the implemented scenarios and networks. We estimate the total estimated simulation time around **300 hours**. We have used two computers to run the simulations: a laptop with a processor of 1.5GHz and 448Mb of RAM memory, and a PC with a processor of 1.72GHz and 512Mb of RAM memory.

Finally, we must mention that all values in the following graphics are represented with their **confidence interval** of 90% (see Annex 3). To do that, we have carried out 5 simulations per point. Shorter confidence intervals are signal of similar results whereas larger confidences intervals mean higher dispersion of values.

8.1. General evaluation of Manhattan scenario

As mentioned before, a complete evaluation of our Manhattan scenario is presented. The simulation consists on a communication between two nodes: node with ID 0 sends CBR (Constant Bit Rate) traffic over UDP at 20kbps rate to node with ID 7. The propagation model is Two Ray Ground in a building environment where a communication only will be established if there is LOS between source and destination node. AODV is the routing protocol selected. The evaluation is carried out considering IEEE 802.11b MAC protocol. Number of nodes increases from 30 to 120 and three speed cases are selected: 5km/h, 35km/h and 65km/h. Eventually, five different transmission ranges are considered from 100 meters to 200 meters. The main parameters of the scenario are listed below:

- Area: 500 m x 500 m
- Nodes: 30 / 60 / 90 / 120
- Speeds: 5km/h / 35km/h / 65km/h
- MAC protocol: IEEE 802.11b and IEEE 802.11p
- Channel: Two-Ray Ground, Two-Ray Ground+LOS, Rician, Rayleigh
- Source traffic: CBR/UDP at 20Kbps

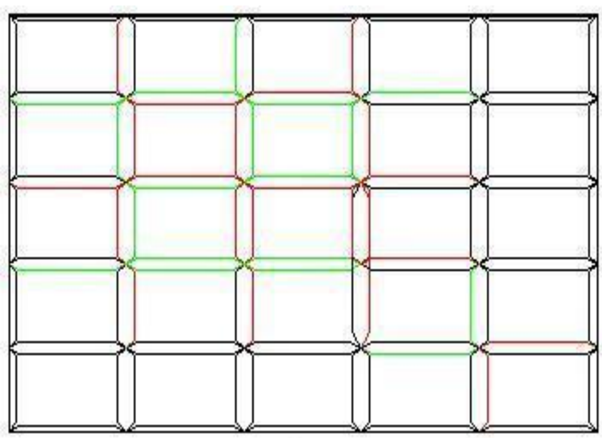


Fig. 8-1 Manhattan scenario

8.1.1. Study of received packet ratio

The first evaluated metric is the ratio of data packets received by destination node (ID 7). For now, the transmission range is set to 100 meters.

8.1.1.1. Range 100 m

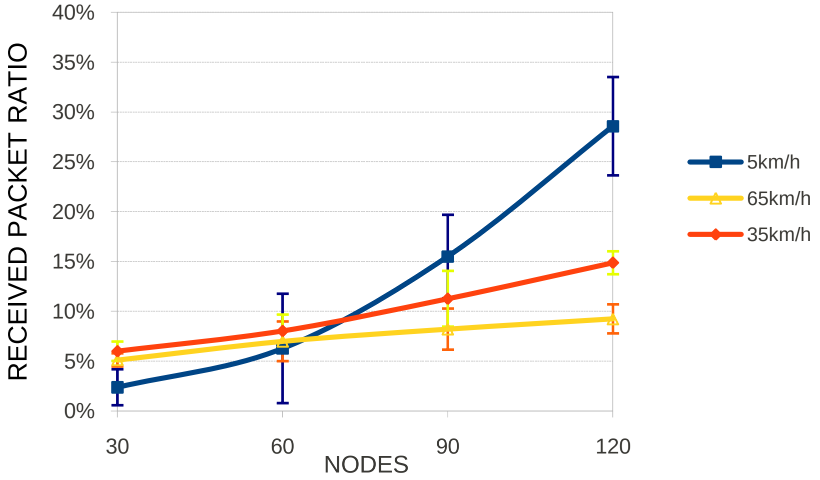


Fig. 8-2 Received packet ratio for range 100 m

Fig 8-2 shows data packet efficiency has an increasing tendency when density of nodes grows whatever the selected speed. Considering 30 nodes, the density of vehicles in the scenario is very low and there is larger distance between nodes than considering a higher density of nodes. Because of that large distance, nodes are more isolated from each other and establishing valid routes is quite unlikely. On the other hand, considering a higher density of nodes allows nodes to be nearer from each other and valid routes are established in an easy way.

Focusing on speed of nodes, when we consider low density of nodes, higher speeds slightly perform better efficiency than lower speeds. However, as density of nodes increases, the situation becomes reversed. Considering a speed of 5km/h quite better efficiency is achieved than the performed by speeds of 35km/h and 65km/h. As stated before, establishing valid routes is more difficult when considering low density of nodes and low speeds because of the distance between the vehicles. This situation can be solved if vehicles begin to move so that this motion gets nodes nearer and allow them to establish valid routes. Therefore, if no speed or low speed is considered, establishing routes becomes more unlikely and low packet efficiency is achieved. On the contrary, when high density of nodes is considered vehicles are closer to each other and establishing routes is an easy process. When a route is established, it should be active as long as possible. In this situation the probability of maintain a valid route is higher considering low speed of nodes than considering high speed of vehicles because routes break often.

From Fig 8-3 to Fig 8-6, how the performance of the simulation is affected by an increase of transmission range is illustrated. As shown in Fig. 8-3, what discussed in the previous analysis of results presented in Fig. 8-2 is still valid. Efficiency gets higher with an increase of number of nodes. Moreover, selecting low speeds performs better efficiency for high density situations contrary to low density environment.

8.1.1.2. Range 125 m

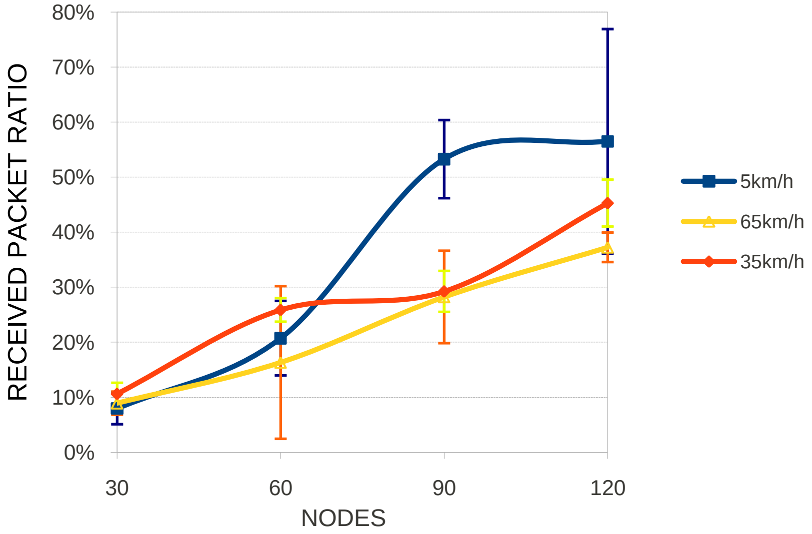


Fig. 8-3 Received packet ratio for range 125 m

However, with a range of 125 meters, the efficiency increases dramatically regardless of speed of nodes or vehicle density. This situation is obvious since now nodes can reach each other easily and route establishment can easily succeed.

The same tendency is shown in Fig. 8-4 to Fig. 8-6 considering transmisison ranges of 150, 175 and 200 meters. However, as the range increases, the packet efficiency for the three speed levels converges with each other. This is because for higher transmission ranges, the coverage time (lifetime of a link between two cars) is higher and the cars' speed has not so high effect in the lifetime of the links.

8.1.1.3. Range 150 m

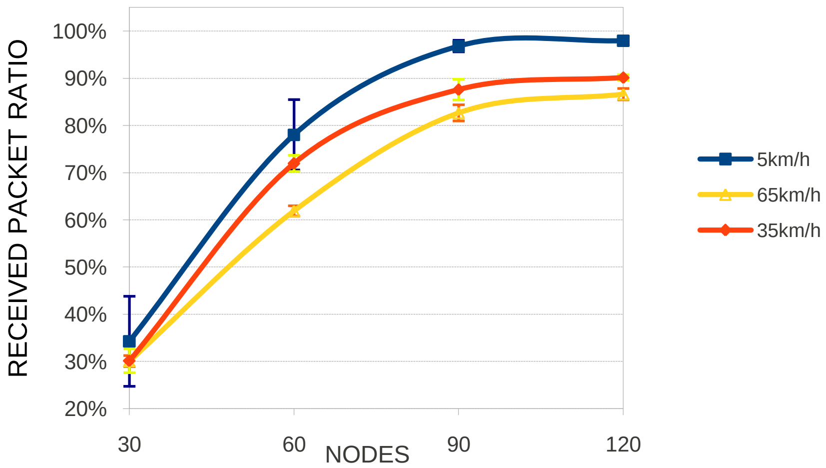


Fig. 8-4 Received packet ratio for range 150 m

In Fig. 8-4 the received packet ratio increases considerably. In addition, the differences between speed levels get shorter. In this situation, where the transmission range has increased significantly, the capacity for nodes to reach each other is higher than considering previous ranges. Therefore, the probability of maintaining routes rises regardless of the selected speed of nodes. However, lower speeds still entail better packet efficiency than higher.

8.1.1.4. Range 175 m

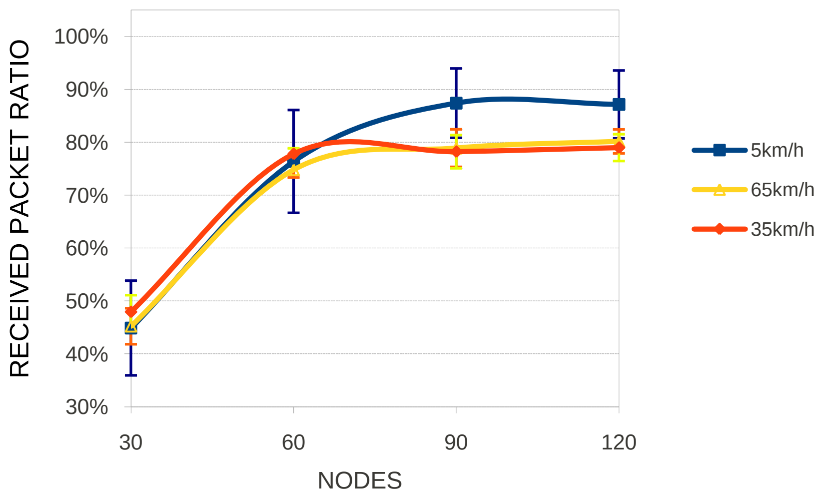


Fig. 8-5 Received packet ratio for range 175 m

Considering Fig 8-5, the enhancement of received packet ratio is confirmed only for low density environment. As transmission range increases from 150 meters to 175 meters, the ratio considering 30 nodes enhances 15-20% regardless the selected speed of nodes. However, the ratio considering 90 and 120 nodes now decreases 10%. This reduction of efficiency is caused by an increase of packet collision. Transmission range is such large that every time a node sent a packet almost all other nodes received that packet. When several nodes are transmitting, the situation leads to a totally use of the shared medium, and thus, the number of collisions increase.

8.1.1.5. Range 200 m

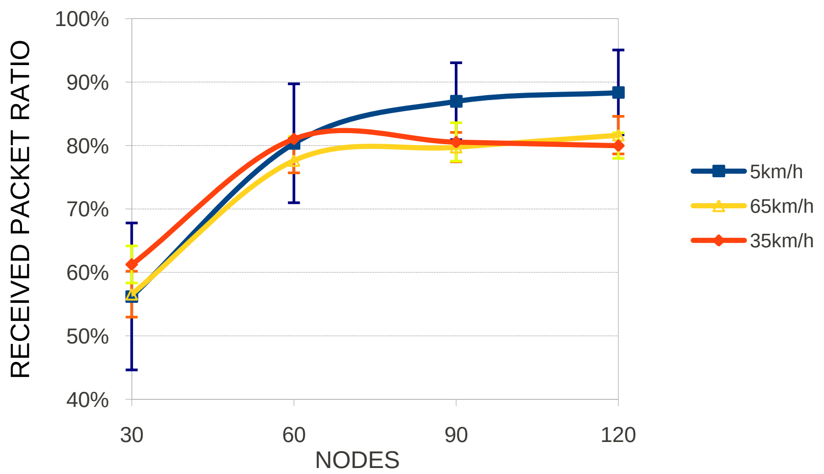


Fig. 8-6 Received packet ratio for range 200 m

It must emphasize that there is total convergence between speed levels selecting low density of nodes. Nevertheless, considering high density of nodes, medium and high speeds entail similar efficiency levels whereas low speed achieves a little better level.

In Fig 8-6, a transmission range of 200 meters is selected. What stated for 175 meters range is still valid. A substancial increase of transmission range (above 150 meters) entails ratio enhancement only in low density environments. Furthermore, it is noticed that medium and high speeds of nodes entail quite similar efficiency levels whereas low speeds achieve an enhancement of 10% over them.

8.1.2. Study of average number of hops

The second evaluated metric is the average number of nodes whereby data packets must hop in order to reach destination node. In the first analysis, a transmission range of 100 meters is considered. Eventually, this range is increased to illustrate how it affects to the simulation performance.

8.1.2.1. Range 100 m

Fig 8-7 shows the average number of hops for a range of 100 meters and considering different speed and density levels. Looking at Fig. 8-7, several issues can be discussed. Firstly, it is considered the effect of density of nodes. If 30 nodes are selected, the average number of hops ranges from 1 to 2 and it increases considering with higher levels of density. That is because, if 30 nodes and a range of 100 meters are considered in the scenario, the probability of reaching each other is quite low. Therefore, only when source node and destination node are greatly closer to each other, a valid route may be established. In this situation the unique routes than can be established may consist of one or two hops. On the other hand, as density of nodes increases, the probability of providing a valid path get higher since there are many intermediate nodes which can perform as relay. Therefore, now routes may consist of various nodes.

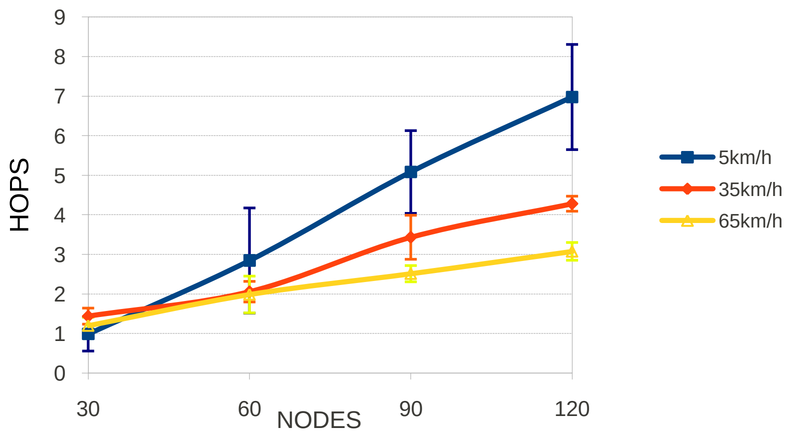


Fig. 8-7 Average number of hops for range 100 m

Now, the effect of node's speed is discussed. From Fig. 8-7 the reached conclusion is that the higher the speed of nodes, the lower the average number of hops. That is because routes composed of large number of nodes are available for less time than routes consisting of a short path of nodes. A route is broken when at least one of the nodes which compose the path changes its position and becomes no longer reachable for its neighbors. Considering nodes moving at high speeds, the possibility of a route composed of two or three nodes to maintain unbreakable for a while may be high. However, the possibility of a route composed of so many nodes to maintain unbreakable for a while is very remote because at least one node will move away. Therefore the current route will be no longer available.

8.1.2.2. Range 125 m

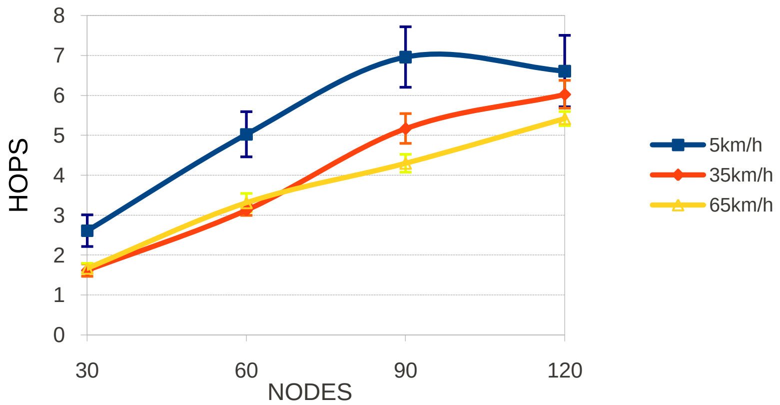


Fig. 8-8 Average number of hops for range 125 m

Fig 8-7 shows the average number of hops for a range of 125 meters and considering different speed and density levels.

Considering a range of 125 meters, the results stated for the previous range are still valid: average number of hops enhances with an increase of density of nodes or with a decrease of the speed of nodes. However, as we see in Fig 8-8 the average number of hops for each speed is slighty higher than the results from Fig 8-7.

8.1.2.3. Range 150 m

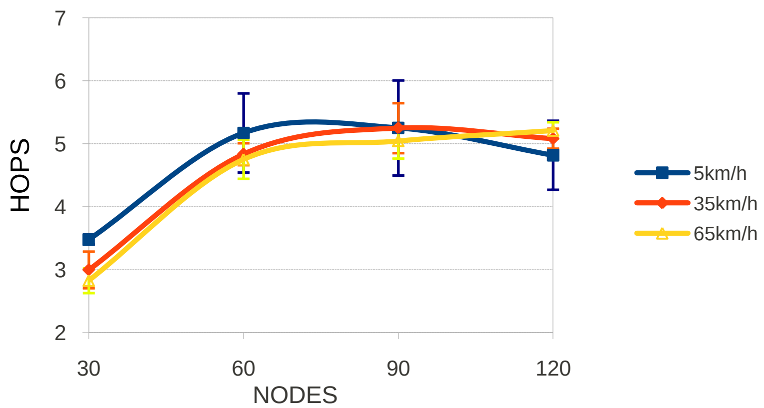


Fig. 8-9 Average number of hops for range 150 m

The enhancement of transmission range entails an increase of the node's reach, which allows nodes to communicate each other despite the distance between them. In this situation, routes than consist of more than one or two hops may be established.

Fig 8-9 to Fig 8-11 shows the average number of hops for transmission range of 150, 175 and 200 meters respectively. As range increases, two tendencies become outstanding. Firstly, the effect of the number of nodes is increasingly less evident. In other words, the average number of hops becomes independent of the selected number of nodes. Now, transmission range is so large that most of nodes can reach each other regardless of the selected density of node in the scenario.

Secondly and as aforementioned, a high enhancement of transmission range leads to more use of the shared medium, and therefore, an increase of packet collision. Now, the level of packet collision is the first reason why the routes are broken instead of the selected speed of nodes. Hence, the average number of hops also becomes independent of the speed of nodes.

8.1.2.4. Range 175 m

Considering transmission ranges of 175 and 200 meters the two tendences descripted above are more evident. The resemblance between the results considering different speed and density of nodes is increasingly obvious. In addition, as in Fig 8-9 the value of average number of hops is around 5, in Fig 8-10 the value has decreased to 4 because of the level of packet collision. In Fig 8-11 the value has slighty decreased to 3.5.

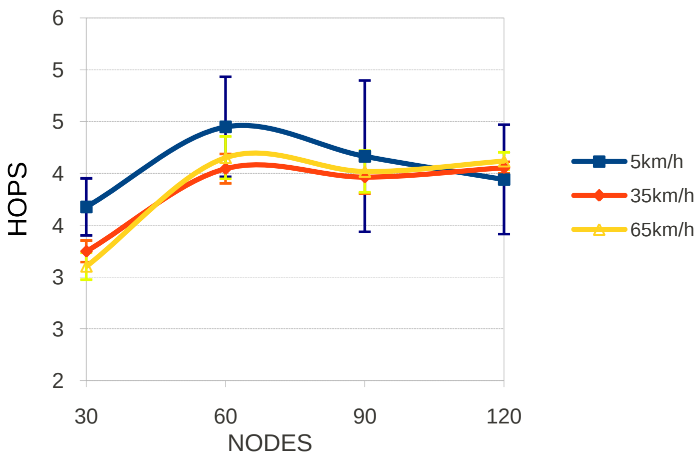


Fig. 8-10 Average number of hops for range 175 m

8.1.2.5. Range 200 m

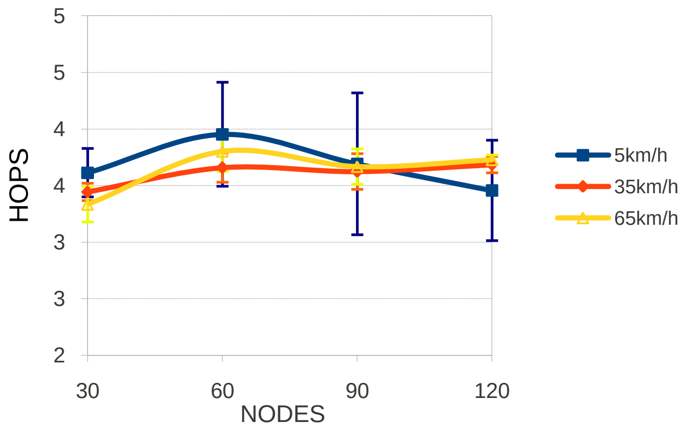


Fig. 8-11 Average number of hops for range 200 m

8.1.3. Study of average end-to-end delay

The following section is aimed to evaluate the average delay between two events: sending data packets by source node and reception of these data packets by destination node.

8.1.3.1. Range 100 m

Fig 8-12 shows the end-to-end delay of data packets considering 100 meters of transmission range and different levels of speed and number of nodes.

The tendency of the average end-to-end delay is utterly connected to the previous evaluation of average number of hops: the longer the established route, the greater the average delay required to go through it. From Fig 8-7 we discussed that for range of 100 meters and considering low density of nodes or high speed of vehicles, the established routes are extremely short. On the other hand, considering high density of nodes or low speeds entails an increase of the length of the path. Both assumptions are consistent with Fig 8-12. Considering 30 or 60 nodes the routes are shorter, and therefore, the value of the average end-to-end delay is under 100ms besides of the selected speed.

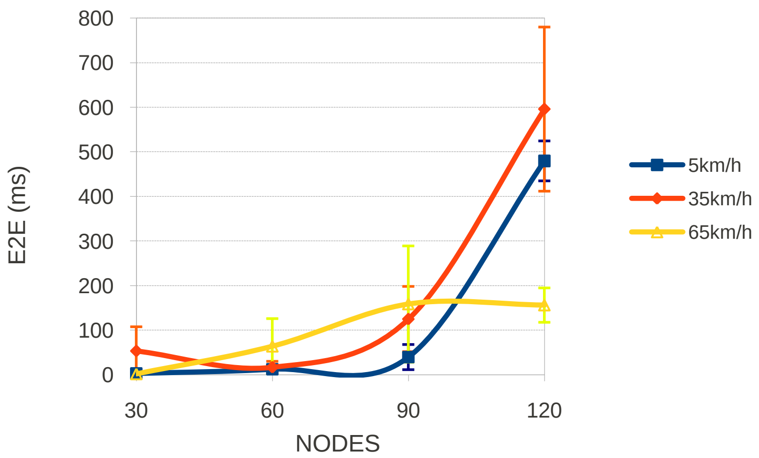


Fig. 8-12 Average end-to-end delay for range 100 m

If we focus on density of 90 or 120 nodes the length of routes enhances considering low and medium speed. Hence, the average value of end-to-end delay increases dramatically. However, in such situation but selecting high speeds of nodes, the average value of end-to-end delay remains under 200ms.

8.1.3.2. Range 125 m

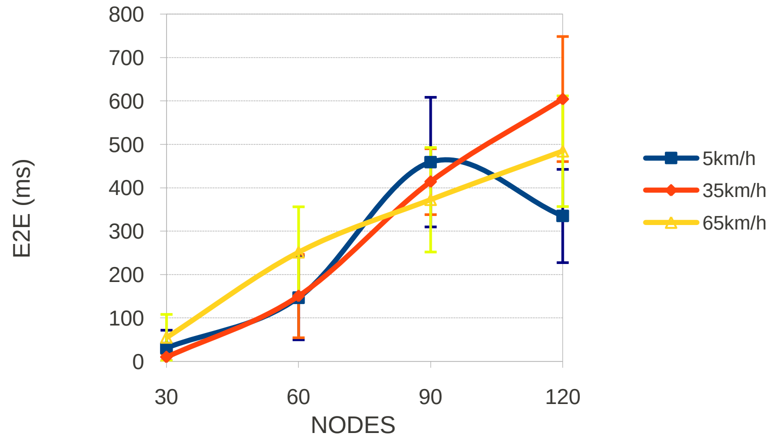


Fig. 8-13 Average end-to-end delay for range 125 m

As stated in the study of average number of hops, the enhancement of transmission range entails an increase of the node's reach. Thus, routes now are composed for a larger number of nodes. Fig 8-13 shows a slight increase of the average end-to-end delay compared to the results in Fig 8-12.

8.1.3.3. Range 150 m

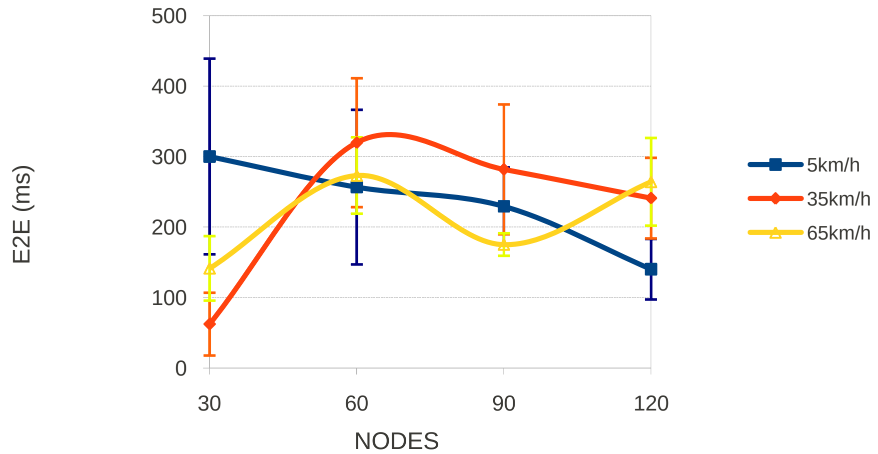


Fig. 8-14 Average end-to-end delay for range 150 m

As discussed in the previous study, the average number of hops becomes more independent of the selected number of nodes as the transmission range increases substancially. Figs 8.14 to Fig 8.16 show the same effect on the average end-to-end delay.

8.1.3.4. Range 170 m

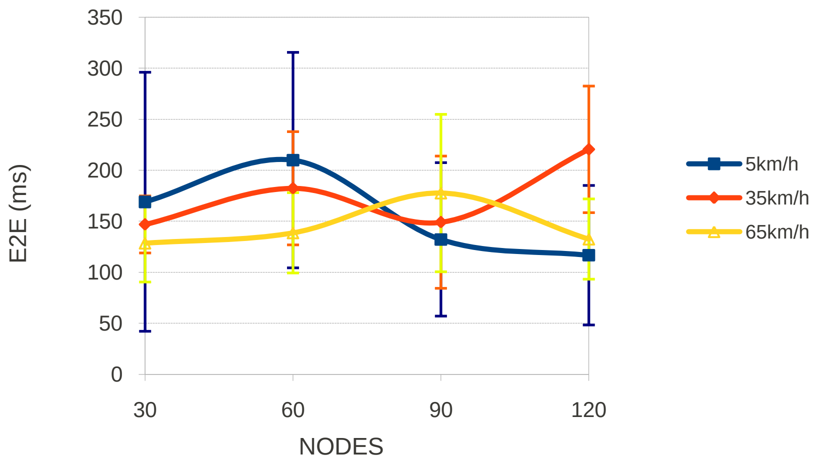


Fig. 8-15 Average end-to-end delay for range 175 m

8.1.3.1. Range 200 m

In addition, we previously discuss how the average number of hops decreases when extremely large transmission range is considered, which is also confirmed for the average end-to-end delay evaluation. In Fig. 8.15 and Fig. 8.16 the average level of end-to-end delay is around 100 and 200 ms, which is under the level considering a range of 150 meters as shown in Fig 8-14.

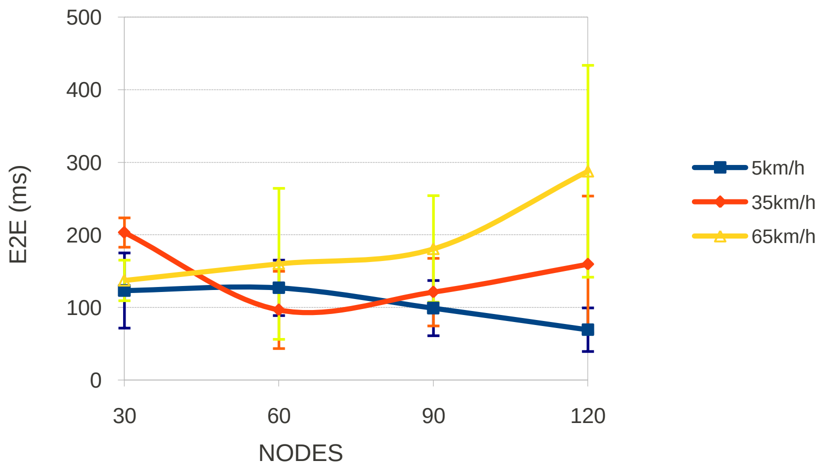


Fig. 8-16 Average end-to-end delay for range 200 m

8.1.4. Study of data throughput

The last evaluated metric is data throughput. Fig. 8-17 to Fig. 8-20 shows the evolution of data throughput throughout the simulation time for different level of node's density and speed. It must be remembered that the source transmit at rate of 20kbps. The following results are highly related to those discussed for the packet efficiency. This makes senses, because the higher the received packet ratio, the higher the data throughput.

8.1.4.1. Range 100 m

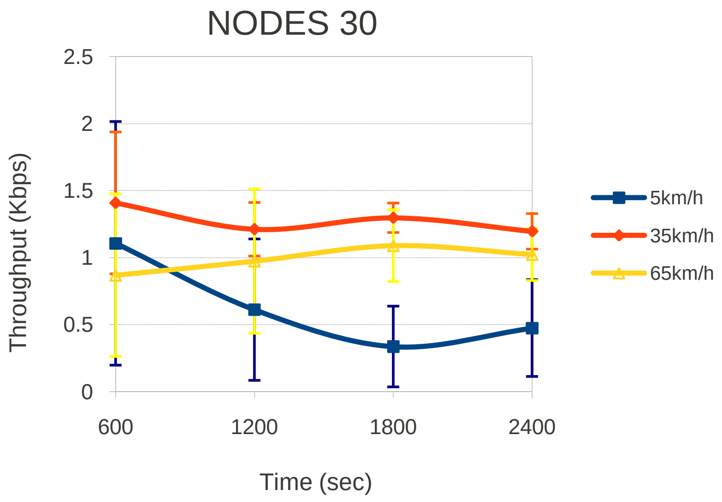


Fig. 8-17 Data throughput for range 100 m and 30 nodes

As stated in the analysis of data packet efficiency in Section 8.1.1., it enhances with an increase of the selected number of nodes. In addition, considering low speeds entails better efficiency for high level of density of nodes. However, for low density of nodes, high speeds are desirable. This behaviour is reflected in Fig 8-17 to Fig. 8-20. For instance, selecting a speed of 5kmh, the value of data throughput is around 0,5kbps for density of 30 nodes whereas considering 120 nodes, the value of data throughput increases around 5-6kbps.

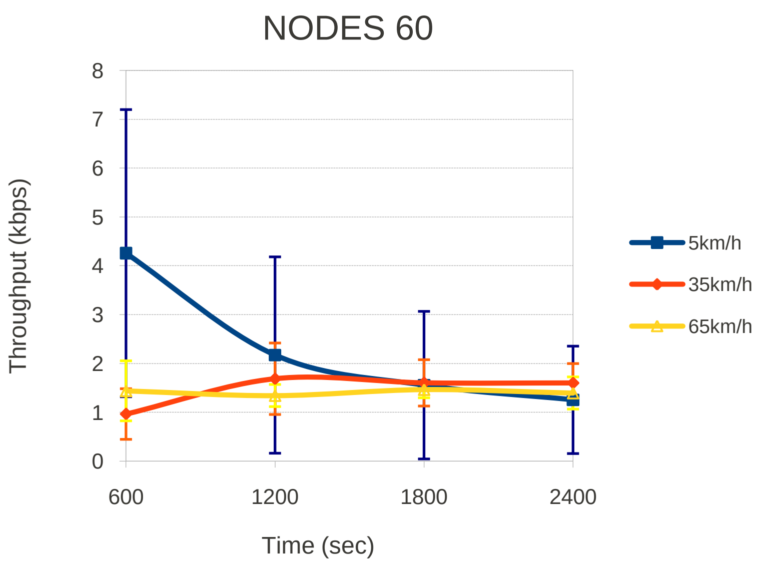


Fig. 8-18 Data throughput for range 100 m and 60 nodes

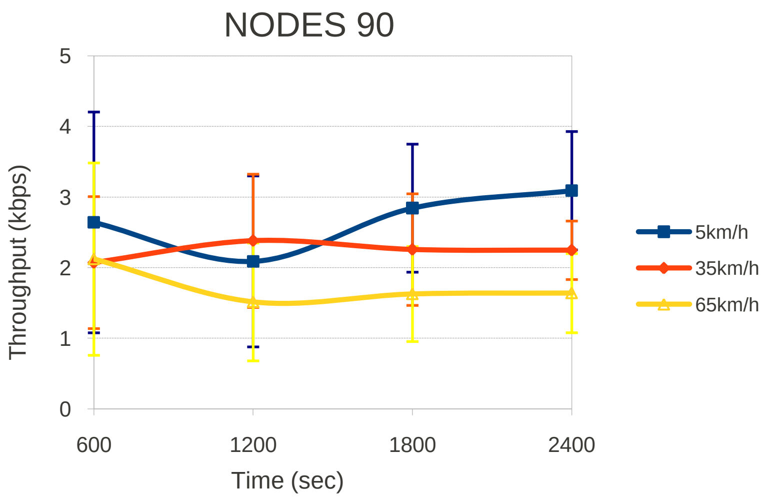


Fig. 8-19 Data throughput for range 100 m and 90 nodes

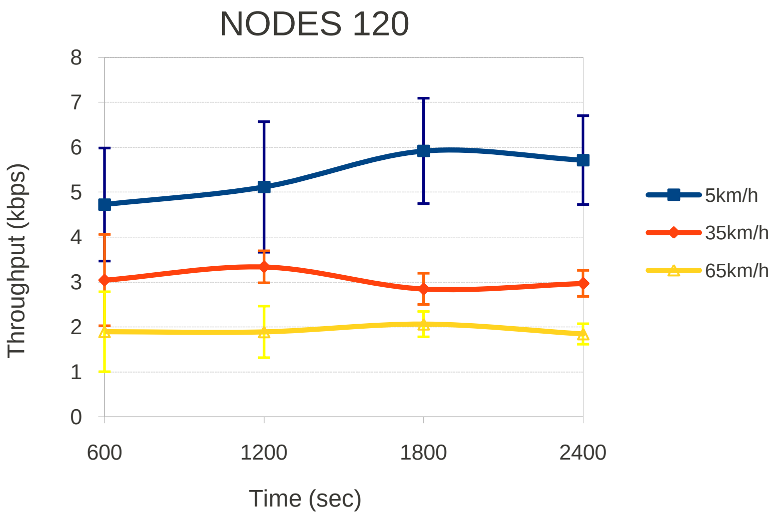


Fig. 8-20 Data throughput for range 100 m and 120 nodes

8.1.4.2. Range 125 m

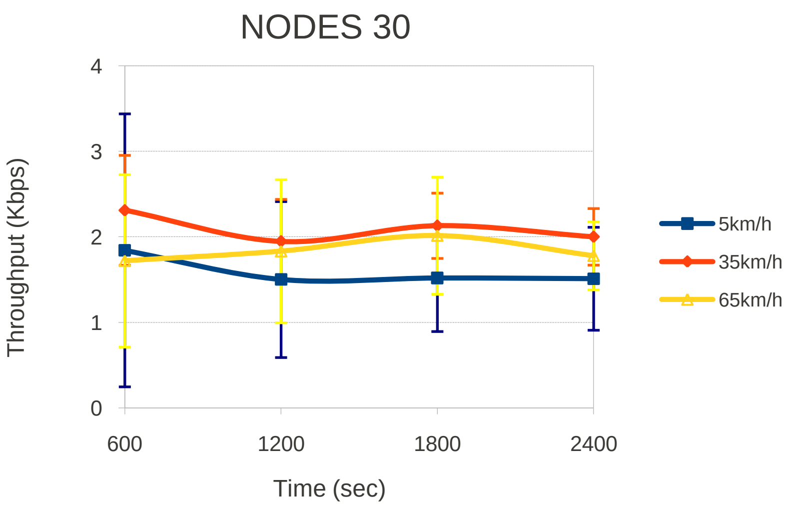


Fig. 8-21 Data throughput for range 125 m and 30 nodes

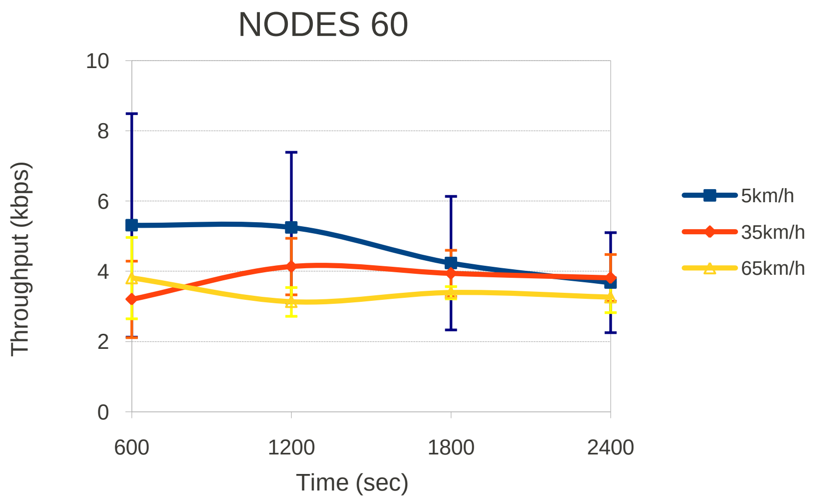


Fig. 8-22 Data throughput for range 125 m and 60 nodes

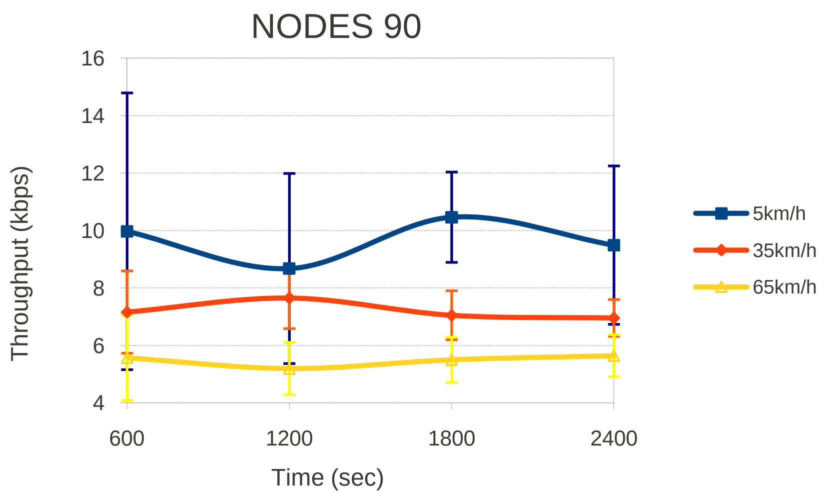


Fig. 8-23 Data throughput for range 125 m and 90 nodes

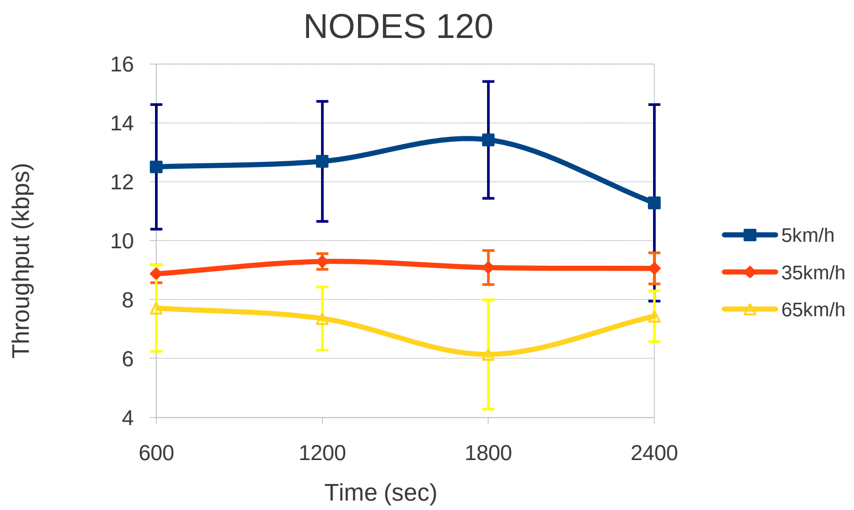


Fig. 8-24 Data throughput for range 125 m and 120 nodes

Considering ranges of 125 meters, the received data packet ratio as well as the data throughput increase dramatically, as shown in Figs 8-21 to Fig 8.24. With this enhance of transmission range and considering 120 nodes and a speed of 5km/h, data throughput value has increased from 5-6kbps to around 12-14kbps. Moreover, as we expected, low speed entails better level of data throughput.

8.1.4.3. Range 150 m

Considering ranges around 150 meters, the enhancement of data throughput as well as the ratio of received packets is obvious. The increase is such substancial that data throughput reaches the optimal value of 20kbps, considering high density of nodes. It must be remembered that the ratio of received packets reaches almost 100% in the same conditons (see Section 8.1.1.3).

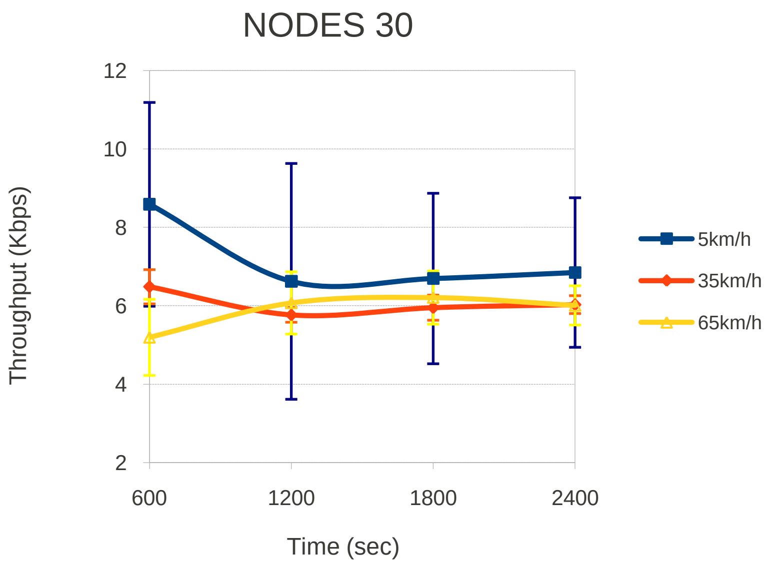


Fig. 8-25 Data throughput for range 150 m and 30 nodes

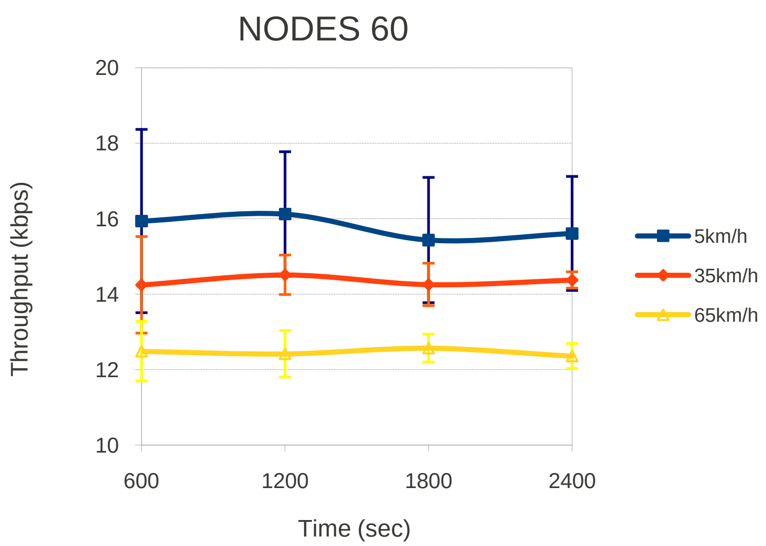


Fig. 8-26 Data throughput for range 150 m and 60 nodes

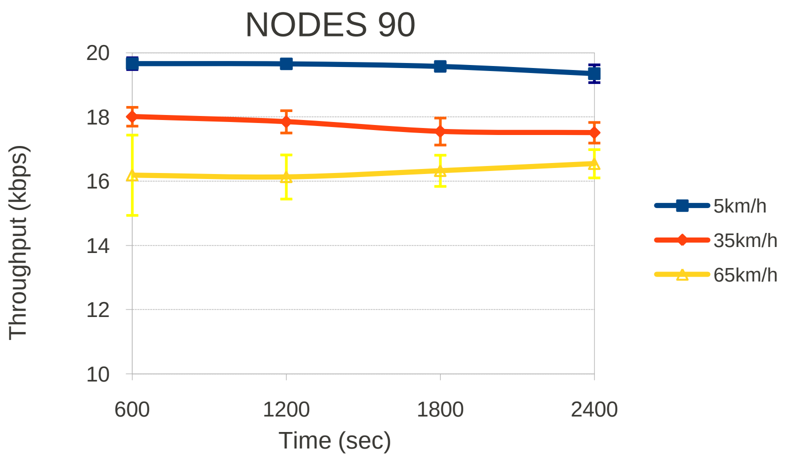


Fig. 8-27 Data throughput for range 150 m and 90 nodes

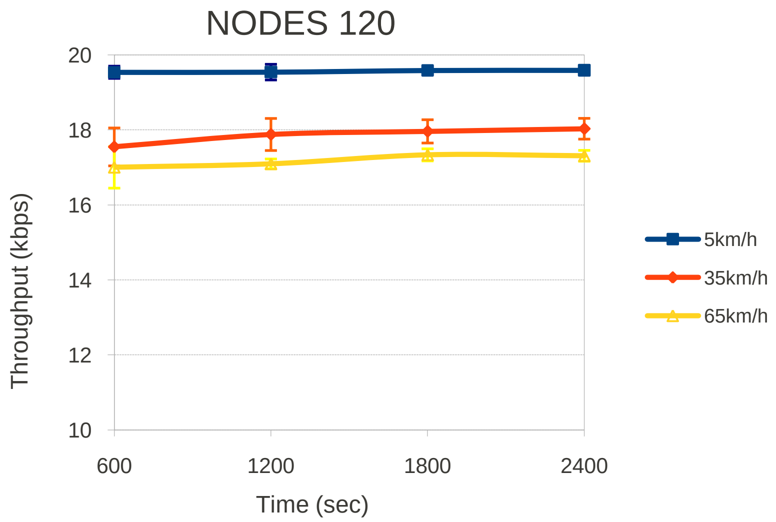


Fig. 8-28 Data throughput for range 150 m and 120 nodes

8.1.4.4. Range 175 m

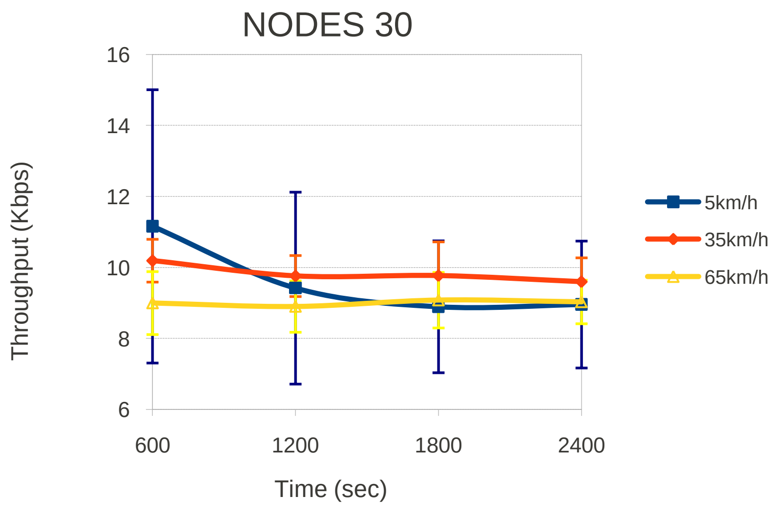


Fig. 8-29 Data throughput for range 175 m and 30 nodes

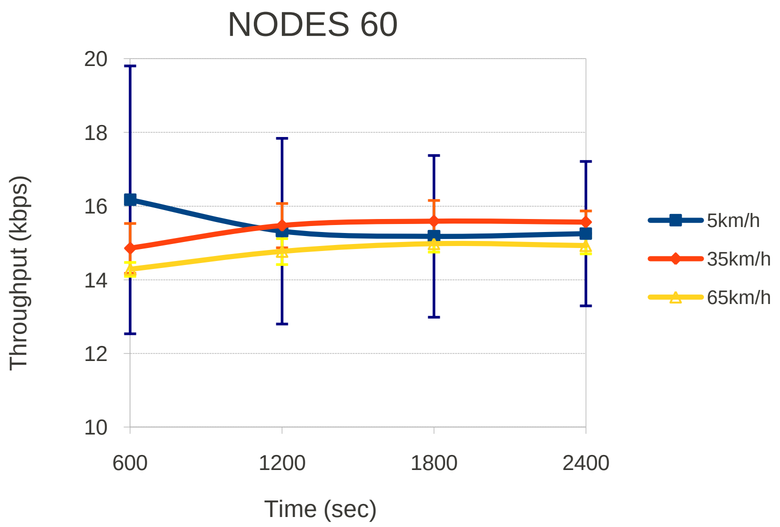


Fig. 8-30 Data throughput for range 175 m and 60 nodes

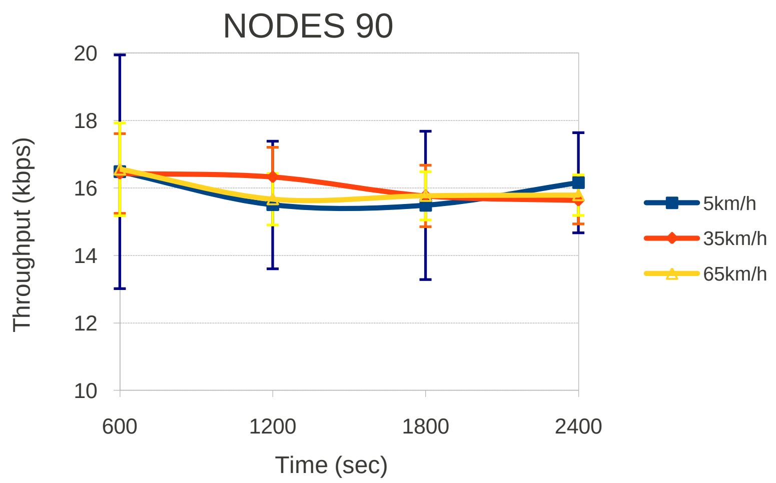


Fig. 8-31 Data throughput for range 175 m and 90 nodes

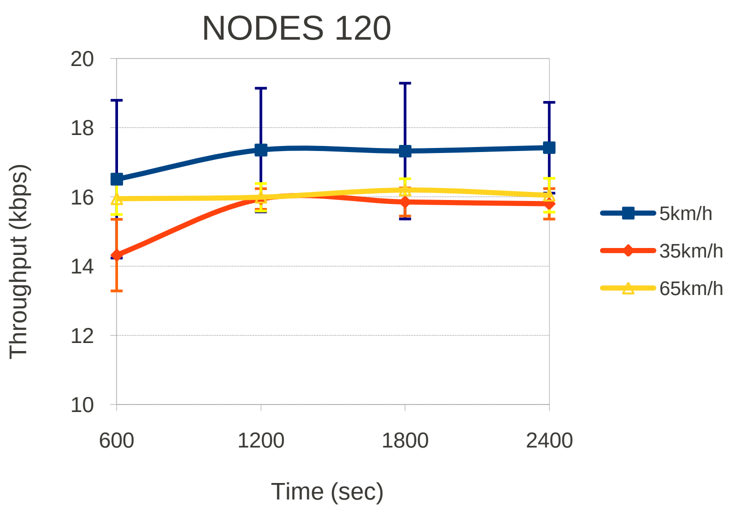


Fig. 8-32 Data throughput for range 175 m and 120 nodes

We should recall what stated in the study of received packet efficiency, where selecting a transmission range over 150 meters entails an enhancement only for low density of nodes: The ratio increased around 15-20% considering 30 nodes but if we select 90 or 120 nodes, the value decreased about 10%. This behaviour is coherent with shown in Fig 8-29 to Fig 8-36, where transmission ranges of 175 meters and 200 meters are considered.

8.1.4.5. Range 200 m

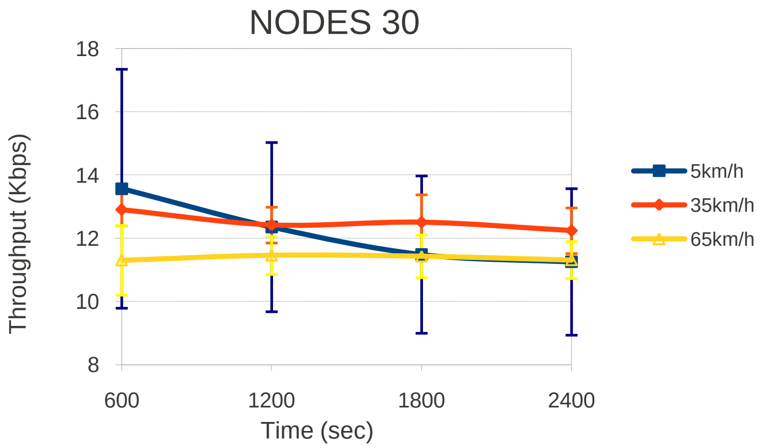


Fig. 8-33 Data throughput for range 200 m and 30 nodes

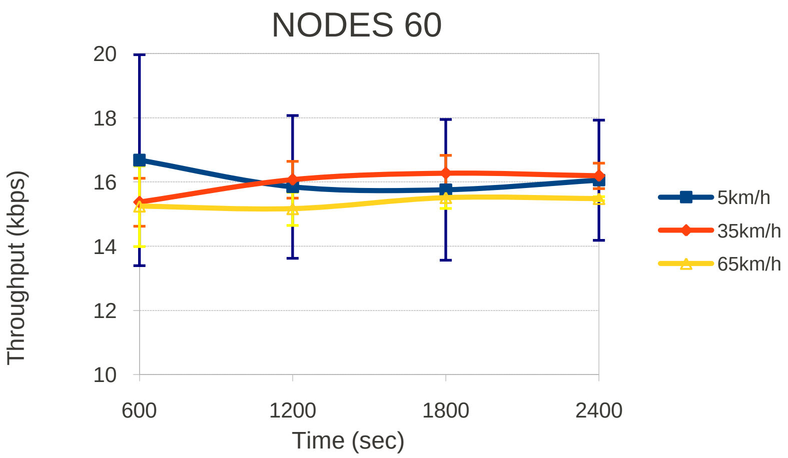


Fig. 8-34 Data throughput for range 200 m and 60 nodes

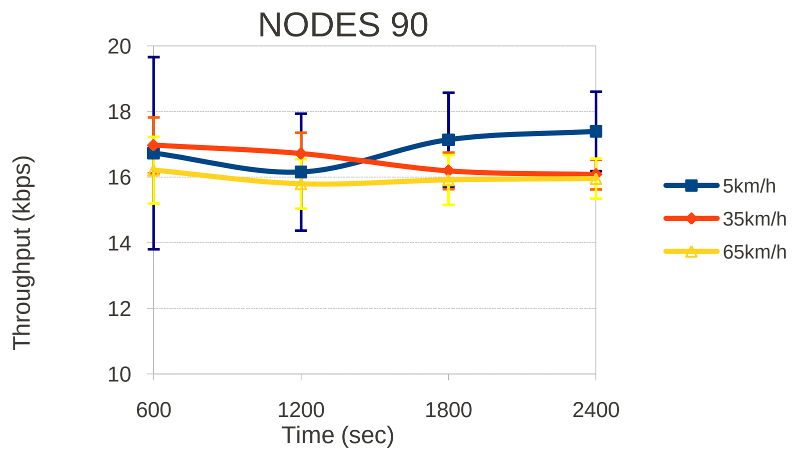


Fig. 8-35 Data throughput for range 200 m and 90 nodes

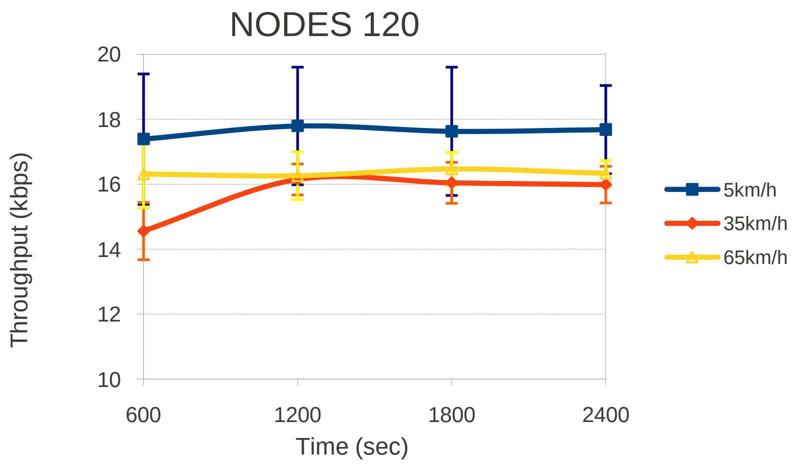


Fig. 8-36 Data throughput for range 200 m and 120 nodes

8.2. Evaluation of MAC protocol

In this section a comparison of the performance evaluation considering IEEE 802.11b and IEEE 802.11p MAC protocols is discussed. The scenario features are still the same presented in the introduction of Section 8.1. The only difference is that now the transmision range and speed of nodes are no longer variable and they are set to 125 meters and 35km/h respectively.

In Fig 8-37 the values of received packet ratio for a transmission range of 125 meters and considering 802.11b and 802.11p MAC protocols are shown. The enhancement considering 802.11p is obvious, since the obtained values increases around 40-50% over the obtained selecting 802.11b MAC protocol. Another curve is shown in Fig 8-37 in order to quantize the performance improvement. This curve corresponds to the results selecting 802.11b MAC protocol but with a transmission range of 200 meters. Both blue curves have the same tendency and take aproximately the same values. Therefore, we can state than, for received packet ratio, the performance of considering 802.11p MAC protocol with a transmission range of 125 meters is quite similar to considering 802.11b MAC protocol with a larger transmission range of 200 meters.

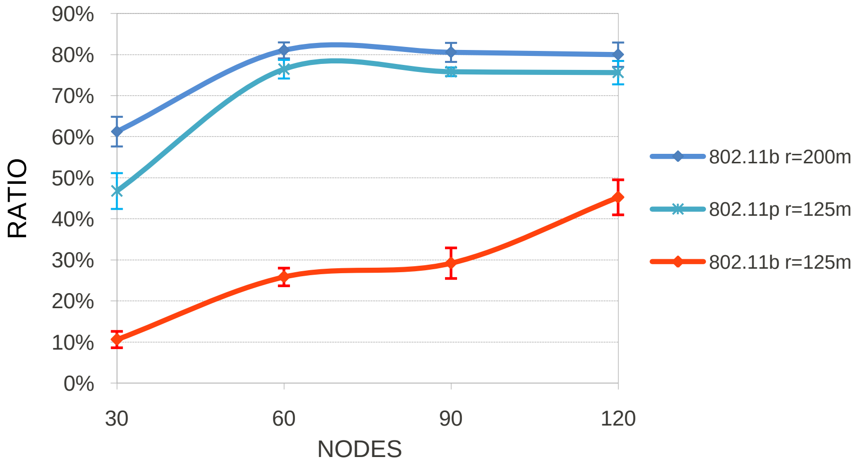


Fig. 8-37 Received packet ratio for different MAC protocols

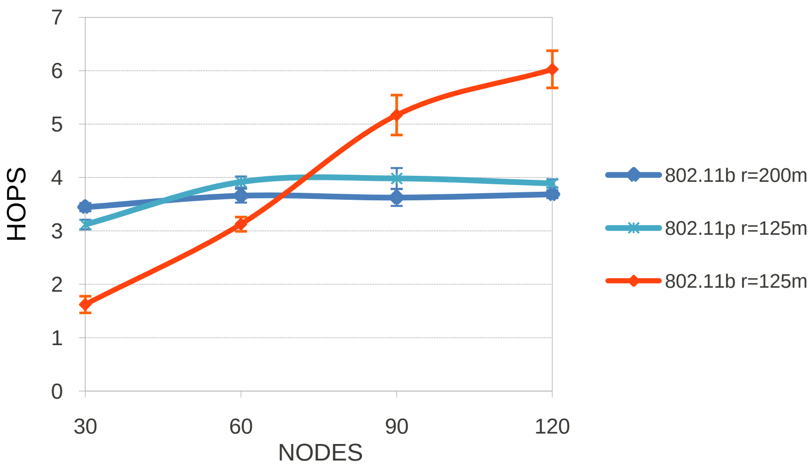


Fig. 8-38 Average number of hops for different MAC protocols

The conclusion stated in the previous analysis is still valid for the average number of hops, as shown in Fig 8-38. The tendency of the average number of hops considering 802.11b MAC protocol and a range of 200 meters, which was discussed in Section 8.1.2.5, is quite close to the tendency of considering 802.11p MAC protocol and a shorter transmission range of 125 meters. However, considering IEEE 802.11p MAC protocol can entail drawback: When low density of nodes are selected (30 or 60 nodes), the average number of hops with the use of IEEE 802.11b is around 2-3, which is under the average number of hops obtained with the use of IEEE 802.11p. This situation reverses considering high number of nodes (90 or 120), where the average number of hops selecting IEEE 802.11b is 1-2 hops greater than the values considering IEEE 802.11p.

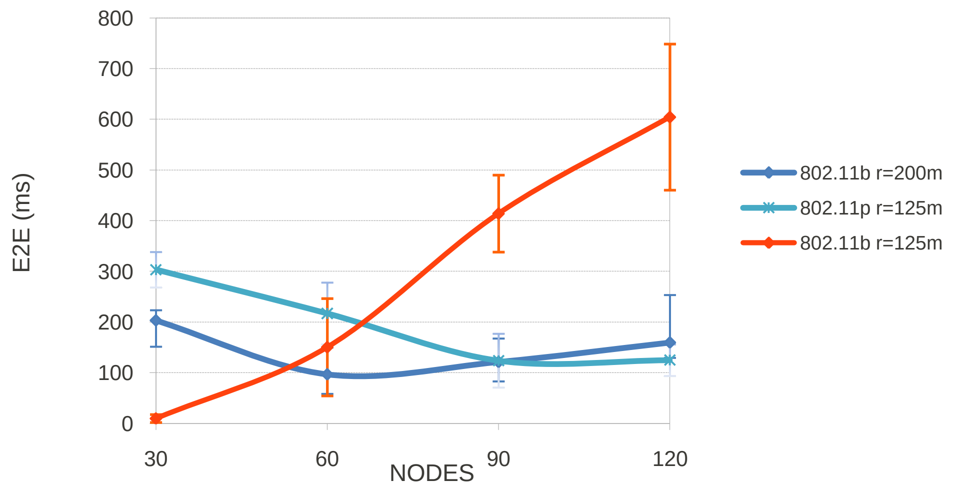


Fig. 8-39 Average end-to-end delay for different MAC protocols

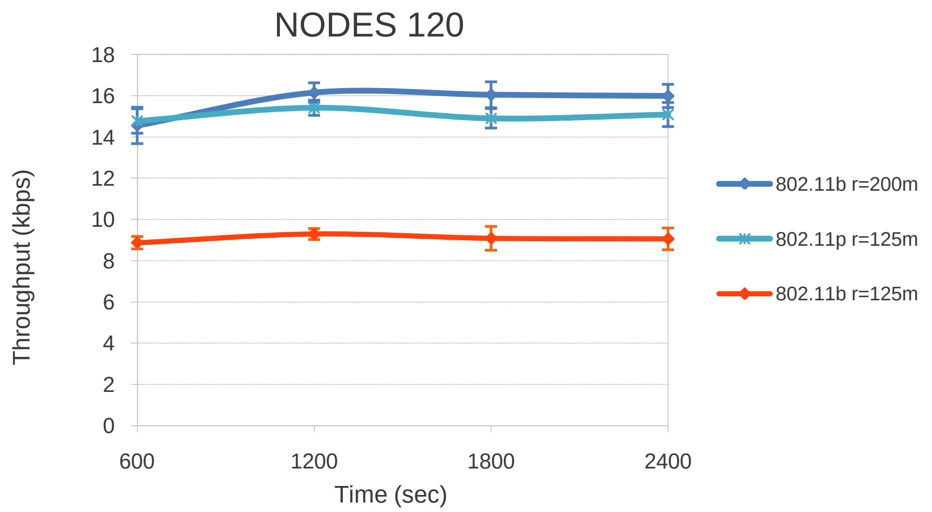


Fig. 8-40 Data throughput for 120 nodes and considering different MAC protocols

In Fig 8-39 the values for the averange end-to-end delay as a function of the density of nodes is shown. The three same curves are illustrated in order to corroborate the network performance considering 802.11p MAC protocol. It also must stated what is just discussed: selecting low density of nodes and considering 802.11b MAC protocol entails lower values of average number of hops, and therefore, lower values of average end-to-end delay.

Fig 8.40 shows the data throughput as a function of time and only considering 120 nodes. The enhancement of data throughput with selecting 802.11p MAC protocol is obvious. The similarities between the two blue curves are also evident.

8.3. Evaluation of propagation model

In this section an evaluation of how the selected propagation model affects the network performance is discussed. Four progation models are proposed: Two-Ray Ground model considering building walls and obstacles where a communication only will be establised if there is LOS between source and destination nodes and original Two-Ray Ground model without considering obstacles. Finally, Ricean and Rayleigh models presented in *Chapter 5* are also proposed.

In addition speed of nodes is set to 35km/h, transmission range is set to 125 meters and 802.11b MAC protocol is used.

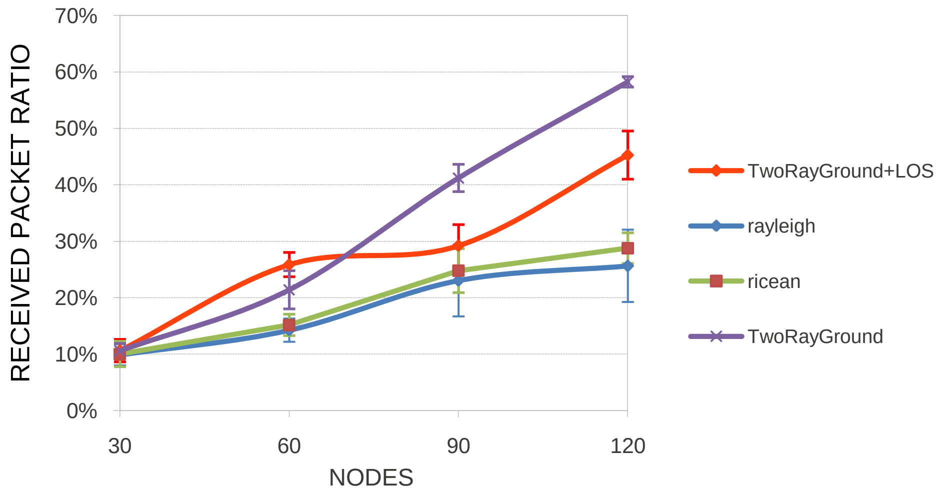


Fig. 8-41 Received packet ratio for different propagation models

A comparison of the received packet ratio considering the four propagation models is shown in Fig 8.41. Two-Ray Ground is the model which offers better received packet ratio, since buildings and obstacles are not considered. Although the probability of establishing routes is quite higher, this model is not well-appropiate for simulating realistic urban scenarios to not consider buildings and obstacles.

The propagation model which offers the second best received packet ratio is Two-Ray Ground with LOS. This is the model which has been considered during previous evaluations and is a better proposal for simulating realistic urban scenarios.

Finally, Ricean and Rayleigh models offers lower received packet ratio because the received signal is affected by fading, which decreases its strength. As stated in *Chapter 5*, Ricean model is more appropiate to simulate realistic urban scenarios because it considers LOS conditions and Rayleigh model does not.

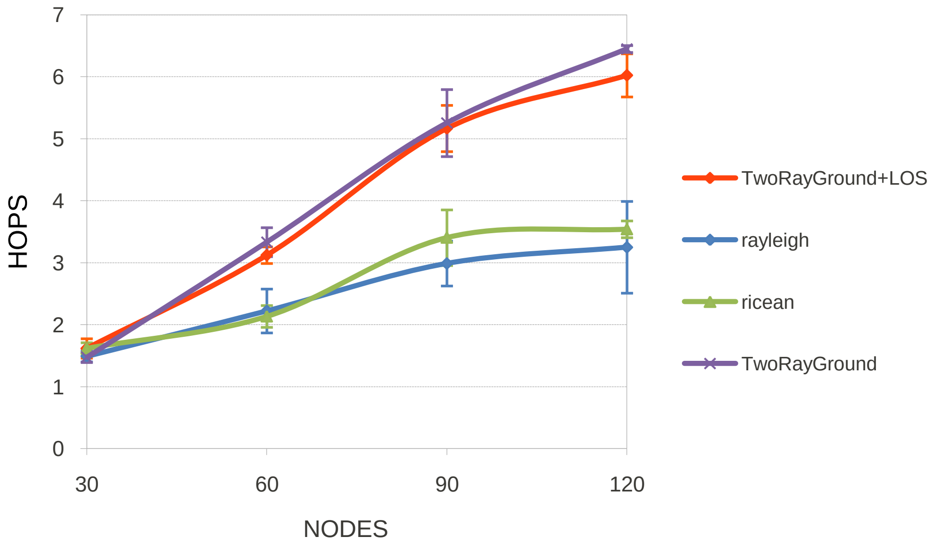


Fig. 8-42 Average number of hops for different propagation models

Despite their differences, both Two-Ray Ground models offer quite similar values in terms of average number of hops. This also occurs between Ricean and Rayleigh model. These two models require low average number of hops since the fading effect drecreases the received signal length and therefore, its reach. As stated in the beginning of the analysis, lower transmission range entails shorter routes and lower number of hops.

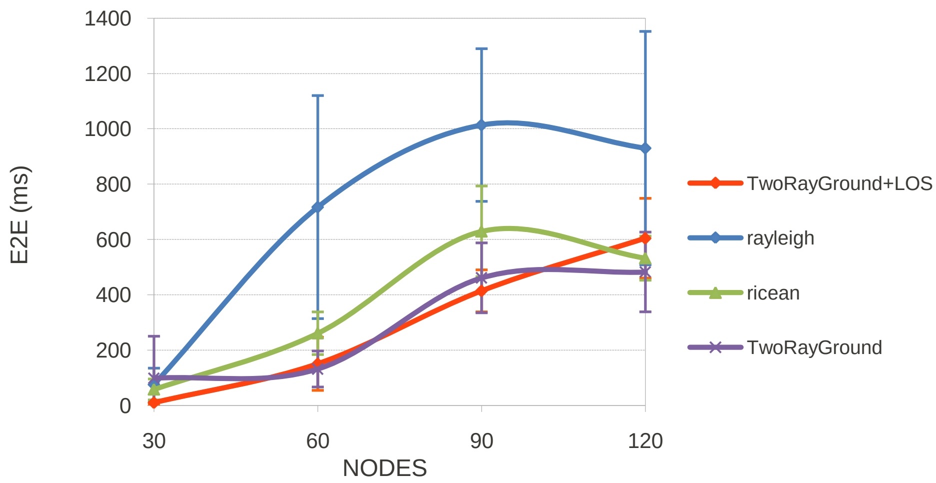


Fig. 8-43 Average end-to-end delay for different propagation models

As for the average end-to-end delay, both Two-Ray Ground models offer quite similar behaviour. The average end-to-end delay for Ricean model, also show similar behavious but its values are slighty higher. Finally, considering the Rayleigh model entails a dramatically increase of the end-to-end delay.

Such as shown in Section 8.1, higher received packet ratio involves higher data throughput. As expected, the original Two-Ray Ground model offers the best values of data throughput and Ricean and Rayleigh models offer the worst.

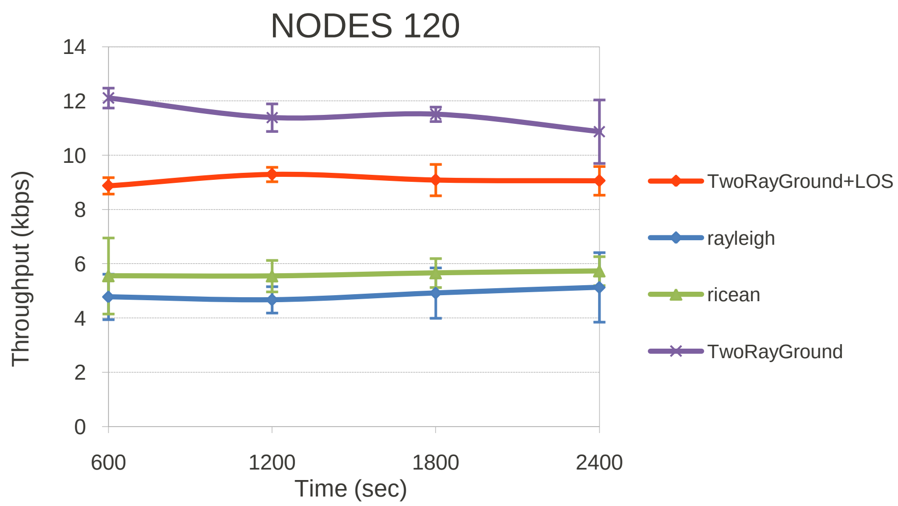


Fig. 8-44 Data throughput for 120 nodes different propagation models

8.4. Evaluation of alternative scenarios

As stated in *Chapter 7*, other three scenarios have been simulated in addition to the Manhattan in order to corroborate the many alternatives offered by VanetMobiSim. These scenarios are two urban scenarios and a highway scenario. Urban TIGER and Highway TIGER have been implemented from real maps available on U.S. Census Bureau. Urban Random is designed with a random and slightly denser street distribution directly with VanetMobiSim. The scenario and network configuration for the two urban scenarios has not been modified, so the transmission range is still 125 meters and the speed of nodes is 35km/h. In addition the propagation model is still Two-Ray Ground with obstacles and LOS requeriments.

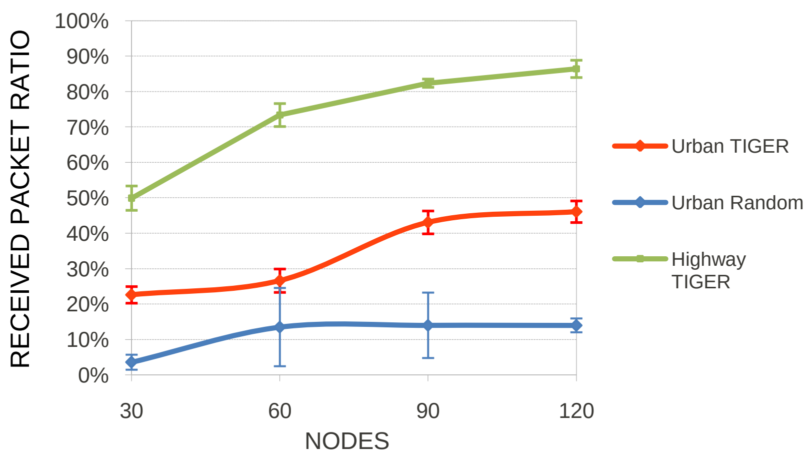


Fig. 8-45 Received packet ratio for alternative scenarios

However highway scenarios has been implement with some difference: the selected speed is within the range 80-120km/h and the propagation model is the original Two-Ray Ground, since considering buildings and walls effect in a highway environment is pointless.

As shown in Fig 8.45, the difference between the three scenarios is obvious. The received packet ratio for the Highway TIGER scenario is quite high contrasting to the other two scenarios for two main reasons. Firsly, the absence of obstacles allows higher number of transmissions. And secondly, the motion and position of nodes is restricted by the highway environment where intersections are not considered, not allowing vehicles to move randomly but one after another.If most of the vehicles are closer and aligned with each other, the probability of maintain valid routes is quite high.

Considering the urban scenarios, received packet efficiency of Urban Random scenario is halfed contrasting to Urban TIGER scenario. This makes sense because the denser street distribution of the streets in Urban Random scenario entails higher number of obstacles, which causes LOS between vehicles is less likely. In this situation, the probability of establishing communications between nodes and maintaining long routes is quite lower.

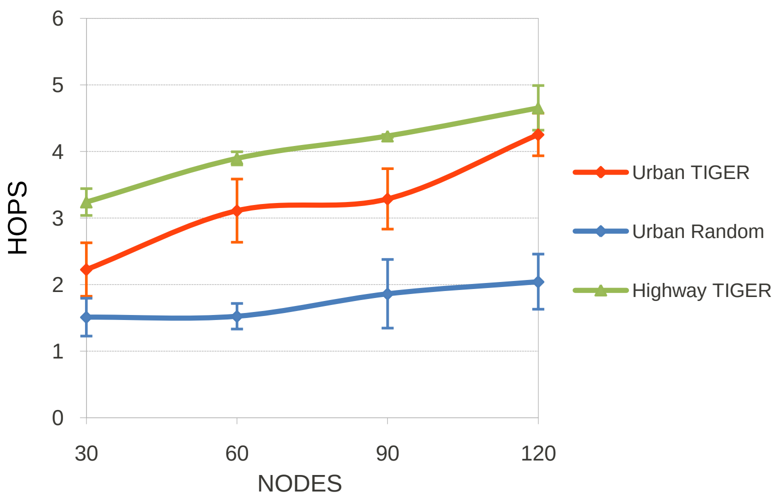


Fig. 8-46 Average number of hops for alternative scenarios

As Urban Random scenario does not allow long routes between nodes, the average number of hops is the lowest as well as the average end-to-end delay. As discussed previously, the position and motion of nodes in Highway TIGER scenario entail longer routes, and therefore, a higher number of nodes and end-to-end delay.

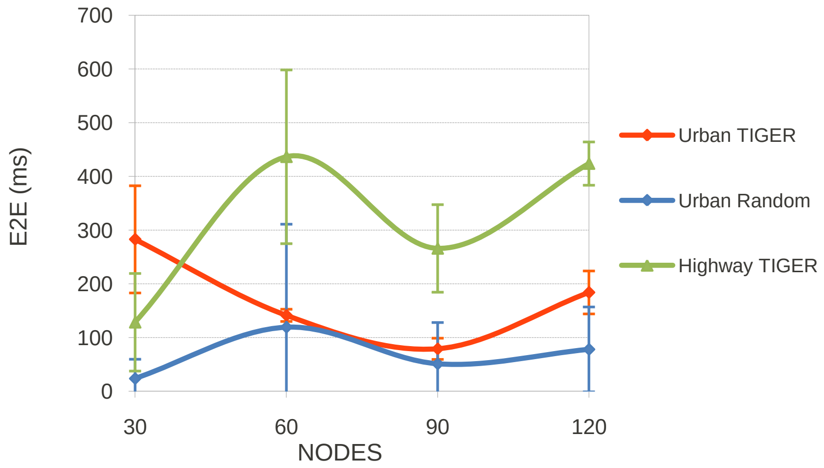


Fig. 8-47 Average end-to-end delay for alternative scenarios

The same tendency is shown in Fig 8-48. Highway TIGER scenario achieves a high level of data throughput whereas urban scenarios entail lower data throughput.

In addition, there is a remarkable issue considering the Urban Random scenario. Data throughput remains around 1-2kbps considering high number of nodes. That is because the density of obstacles is such high that if 120 nodes are selected, the probability of establishing routes is still low. Hence, a major number of nodes should be required in order to handle the obstacles and achieve an enhancement of data throughput.

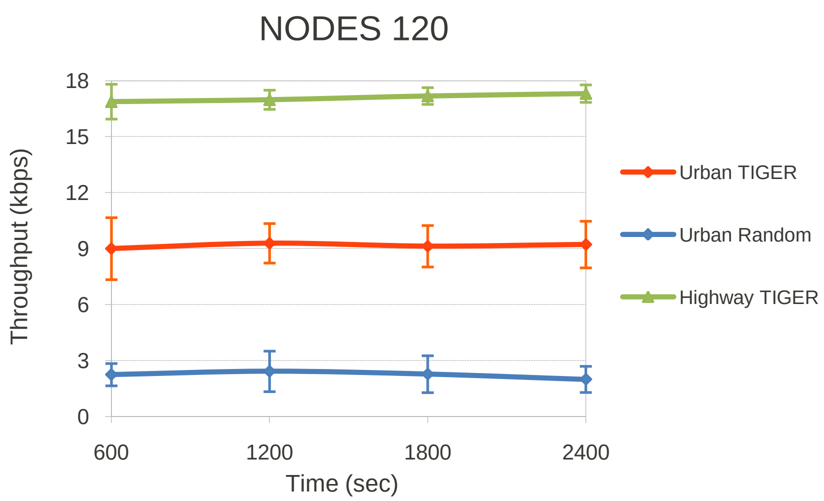


Fig. 8-48 Data throughput for 120 nodes and alternative scenarios

Finally, the most outstanding results that we have discussed throughout this chapter are listed before:

- The received packet ratio and data throughput are better when we select high number of nodes.
- The received packet ratio and data throughput also enhance as transmission range increases untill 150 meters. An increase of 150 m to higher transmission range (175 m or 200 m), both received paket ratio and data throughput decreases because of the number of packet collisions.
- Higher speeds leads to worse performane of received packet ratio and data throughput comapring to lower speeds because of routes break more frequently and the routing protocol has to search new routes more often.
- Average end-to-end delay and number of hops are higher if the number of nodes increases. However, both metrics are lower with slow speeds and short transmission range.
- Considering the MAC protocol, IEEE 802.11p performances better efficiency in terms of received packet ratio and data throughput.
- Furthermore, the enhancement is also observed for the average number of hops and end-to-end delay, whenever we select medium and high number of nodes.
- Among the four implemented propagation models, the original Two-Ray Ground models achieve better performance in terms of received packet ratio, data throughput and end-to-end delay, followed by the Two-Ray Ground with buildings model.
- The performances considering Rician and Rayleigh models are the worst. All these assumptions concorde with what is expected.
- However, Rician model and Two-Ray Ground with obstacles are adequate for VANET enviroments: Rayleigh model is only appropiate in non-LOS conditions and the original Two-Ray Ground does not consider reflections on buildings and walls..

9. CONCLUSIONS AND FUTURE WORK

This project aimed to study and evaluate the performance of vehicular ad hoc networks (VANETs). In this work, the most outstaning mobility models and generators for VANETs were studied. After testing many mobility models generators, we concluded that **VanetMobiSim [80] generator coupled with NS2 [63] network simulator** become the most complete VANET simulator, allowing us to recreate and evaluate practically any vehicular scenario.

Several scenarios and hundreds of simulations were run. We studied the influency of several configuration parameters such as the number and speed of vehicles, the transmission range, the propagation model used and the selected Medium Access Protocol (MAC) protocol. To evaluate the efficiency of the scenarios, four metrics have been evaluated: **received packet ratio, data throughput, average number of hops and end-to-end delay**. After the simulation process, the obtained results are described below.

9.1. Conclusions

The received packet ratio and data throughput enhance with an increase of the number of nodes. In addition, higher speeds leads to worse performance comparing to lower speeds, since links and routes break more frequently and in this case the routing protocol has to recalculate routes periodically. We also observed that an increase of the transmission range from 100 or 125 to 150 meters makes a better performance in terms of received packet ratio and data throughtput, reaching almost an efficiency of 100%. For higher values of transmission range, both metrics become poorer because of the increase of packet collisions. Therefore, 150 meters is the optimal transmission range to achieve a best performance in terms of received packet ratio and data throughput.

In the same way, average end-to-end delay and number of hops increase with the number of nodes and decrease with the speed of nodes and the transmission range, due to the high number of packet collisions and the short lifetime of routes.

Analyzing the four evaluated metrics mentioned before, we noticed that as the transmission range increases, the result always will be similar independently of the values of number of nodes and average speed.

Regarding the two MAC protocols implemented, IEEE 802.11b and IEE 802.11p, the latter dramatically performances better received packet ratio and data throughput. This better performance can be also concluded for average number of hops and end-to-end delay, since the obtained values for IEEE 802.11p are lesser than the values obtained for IEEE 802.11b. This afirmation can be asserted whenever the selected number of nodes is not excessively low. This expected behaviour is similar to what we have explained in chapter 3.

Regarding the four propagation models implemented, we will clasify the results for the four models in terms of data throughput, end-to-end delay and received packet ratio from the best performance to the worst performance as the following: The original **Two-Ray ground model**, **Two-Ray ground with obstacle** environment model, **Rician fading model** and **Rayleigh model**. As expected, the original Two-Ray ground has the best performance and the Rayleigh has the worst one. However, only Rician model and Two-Ray ground with obstacles model are well-appropiate to evaluate realistic vehicular environmet, since Rayleigh model is considered to be used only in non-LOS conditions and the original Two-Ray ground model does not consider buildings and walls.

9.2. Future works

AODV (Ad hoc On-demand Distance Vector) was the routing protocol used during the simulation process, which is not the best routing protocol to handle the high mobility of nodes and short duration of routes present in VANETs. A proposal as a future work could be the evaluation of existing routing protocol which considers car position, trajectories or speeds gathered via GPS that could lead to better results. The Doctorate Thesis by Ahmad Mezher tackles this issue.

In this project, the implemented network is only based on car-to-car communications in urban environment. A proposal could be the implementation of a complex network that could combine not only car-to-car comunication but also car-to-infrastructure comunication. The systems could be a hybrid sensor-vehicular network consisting on the communications between vehicles driving on a highway environment and a network of sensors or roadside infrastructure. Also, this issue is going to be dealed in the Thesis of Mezher.

In addition, our VANET implementation and results may be used as a scalable platform so that current and future researchers on the Department of Telematics Engineering can help them to develop their PhD thesis. The evaluation of video-streaming over VANET and its performance could be a very interesting example.

Finally, we are finishing an article with our interesting results obtained in this project that we will submit to the JITEL (Jornadas de Ingeniería Telemática) congress in spring 2012. A proposed outline of the article is presented in Annex 5.

ANNEX 1. HOW TO RUN VANETMOBISIM AND NS2.

In this section a step by step review of the simulation process is described in order to illustrate how to use every tool properly.

In the following, a real example will aid to review the entire simulation process from VanetMobiSim scenario defination to AWK file filtering.

A1.1. VanetMobiSim

In this first section a scenario must be defined in a XML file. The scenario proposed defines 10 nodes, a simulation area of 200x200 meters, a simulation time of 400 seconds and a Random spatial model:

```
<?xml version="1.0"?>
<!-- Cars in a City with downtown area using the SpaceGraph -->
<universe>
  <dimx>200.0</dimx>
  <dimy>200.0</dimy>
  <!-- <seed>8</seed> -->
  <extension class="de.uni_stuttgart.informatik.canu.mobisim.extensions.NSOutput" output="test_trace"/>
  <extension class="de.uni_stuttgart.informatik.canu.mobisim.simulations.TimeSimulation" param="400.0"/>

<!--Spatial environment -->
  <extension name="SpatialModel" class="de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel" min_x="0" max_x="200" min_y="0" max_y="200">
              <max_traffic_lights>3</max_traffic_lights> <!--specifies the number of intersections managed by traffic lights-->
              <reflect_directions>false</reflect_directions> <!--specifies if the spatial model physically differentiates the two traffic flows-->
              <number_lane full="true" dir="false">1</number_lane>   <!--specifies the number and characteristics of multi-lane roads-->

                                                                      <!-- full – specifies whether all roads have multiple lanes or not-->

                                                                      <!-- max – if the <full> attribute is false, specifies the maximum number
                                                                       of roads with multi-lane-->

                                                                      <!-- dir – specifies if the spatial model physically differentiates the
                                                                      two traffic flows. If the <full> attribute is true, <dir> and
                                                                      <reflect_directions> are equivalent. If not, <dir> allows the user to
                                                                      differentiate the directional flows of multi-lane roads only-->
        </extension>


<!-- Traffic Light-->
<extension name="NewTrafficLight" class="eurecom.spatialmodel.extensions.TrafficLight" spatial_model="SpatialModel" step="10000"/> <!-- step in ms-->


<!-- Spacegraph -->
  <extension class="eurecom.spacegraph.SpaceGraph"  cluster="true">
        <clusters density="0.0001"> <!-- 4 cluster in 40.000m² -->
                <cluster id="downtown">
                       <!-- <density>0.0025</density> -->
                       <density>0.003</density>  <!-- 30 obstacles per 10.000m² (cluster area) -->
                              <ratio>0.25</ratio>
                              <speed>2.77</speed>  <!--in m/s-->
                      </cluster>
                <cluster id="suburban">
                      <!-- <density>0.0001</density> -->
                       <density>0.001</density> <!-- 10 obstacles per 10.000m² (cluster area) -->
                              <ratio>0.75</ratio>
                              <speed>4.16</speed> <!-- in m/s-->
                      </cluster>
                </clusters>
  </extension>

  <extension name="PosGen" class="de.uni_stuttgart.informatik.canu.tripmodel.generators.RandomInitialPositionGenerator"/>
  <extension name="TripGen" class="de.uni_stuttgart.informatik.canu.tripmodel.generators.RandomTripGenerator" >
       <reflect_directions>false</reflect_directions>
               <minstay>10</minstay> <maxstay>50</maxstay>
      </extension>
```

```xml
<!-- Nodes definition-->
  <nodegroup n="10">
    <extension class="polito.uomm.IDM_LC" initposgenerator="PosGen" tripgenerator="TripGen">
      <minspeed>2</minspeed>            <!--  7.2km/h-->
      <maxspeed>5</maxspeed>            <!--  18km/h-->
      <step>20</step>
      <a>0.2</a> <!-- acceleration-->
      <b>0.7</b> <!-- deceleration-->
    </extension>
  </nodegroup>

<!-- GUI definition-->
        <extension class="de.uni_stuttgart.informatik.canu.mobisimadd.extensions.GUI">
           <width>640</width>
           <height>480</height>
           <step>1</step>
        </extension>
        <!--
        <extension class="de.uni_stuttgart.informatik.canu.spatialmodel.extensions.DumpSpatialModel" output="dumped_graph.fig"/>
        -->
</universe>
```

Fig. A1-1 VanetMobiSim XML file

All fields and specifications are descripted below:

Area dimensions, seed, simulation time and output trace file are included in this section. With <dimx> tag and <dimy> tag the x-dimension and y-dimension of the simulation area in meters can be specified (notice they only can be used in scenarios with rectangular-bounded simulation areas). With the <seed> tag the random number generation used by VanetMobiSim can be specified.

An instance of global extension is added using the <extension> tag and with <name> tag the name of the class to be instantiated is defined. With an instance of class *de.uni_stuttgart.informatik.canu.mobisim.extensions.NSOutput* a trace file with NS2 format can be defined. Time simulation can be also be defined with an instance of class *de.uni_ stuttgart.informatik.canu.mobisim.simulations.TimeSimulation*.

- Spatial environment

A spatial environment is added with an instance of *de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel*. The spatial environment extension adds support for multilane or multiflow roads and traffic lights at intersections. The XML file proposed includes comments (in <!-- -->) clarifying the use of different tags.

- Traffic lights

Support for traffic lights at intersections can be added using an instance of the *eurecom.spatialmodel.extensions.TrafficLight* extension, allowing definition of the time interval between traffic light changes in *ms*.

- Space Graph

As mentioned in *Chapter 6*, the spatial model can be created from four different ways. The proposed XML file has been implemented with the *Random* option, also known as Space Graph. A Space Graph is added with an instance of *eurecom.spacegraph.SpaceGraph* extension. This creates a random graph built by applying a Voronoi tessellation to a set of randomly distributed points (obstacles).It is possible to define areas or clusters with different obstacles densities, creating a non-homogeneous graph. The characteristics of the clustering are specified using the <cluster> tag. In the proposed scenario the clustering density value is *0,0001clusters/m*^{2}, which means that the simulation area of 40*.000m*^{2} is divided in 4 clusters. Then two kinds of areas have been defined in the proposed scenario: a downtown area with a density value of *0,003 obstacles/m*^{2} and a suburban area with a density value of *0,001 obstacles/m*^{2}. The <ratio> tag specifies the percentage of the kind of cluster in the simulation area, the <speed> tag specifies the maximum speed in *m/s* allowed on the road segments created with this culuster type and the <density> tag specifies the density of obstacles in the cluster in *obstacles/m*^{2}.

- Initial position and trip generator

A random generation of the initial position of nodes and their trip model during the simulation can be defined with an instance of *de.uni.stuttgart_informatik.canu.tripmodel.generators.RandomInitialPositionGenerator* and *de.uni_stuttgart.informatik.canu.tripmodel.generators.RandomTripGenerator* extensions respectively.

- Node Definition

Multiple nodes are added to the simulation using the <nodegroup> tag. To simulate node's motion using the IDM_LC an instance of *polito.uomm.IDM_LC* extension is used. Vehicles moving according to the IDM_LC model support smart intersection management: they slow down and stop at intersections, or act according to traffic lights, if present. Also, vehicles are able to change lane and perform overtakings in presence of multi-lane roads.

Although in the proposed scenario only include a few specifications such as acceleration or maximum or minimum speed, the VanetMobiSim User's Guide [80] includes an extensive list of features than can be configure.

- GUI definition

During simulation a GUI appears displaying the Spatial Model and its elements such as roads and nodes motion. With an instance of *de.uni_stuttgart.informatik.canu.mobisimadd.extensions.GUI* extension, GUI's width and height can be select. Moreover a frame of the Spatial Model can be saved to an output file in *.fig* format with an instance of the *de.uni_suttgart.informatik.canu.spatialmodel.extensions.DumpSpatialModel* extension.

For further descriptions of any tag or specification, the VanetMobiSim User's Guide is available [80].

Once the scenario XML file is configured, the next step will be run VanetMobiSim. To launch the framework, it must change to the directory with framework's files and type:

```
roger@roger-laptop:~/NS2/VanetMobiSim/jar$ java -jar VanetMobiSim.jar scenario.xml
```

Fig. A1-2 VanetMobiSim shell launching

When simulation ends, a file named *test_trace* has been created. It contains the mobility traces for NS2 simulation.

A1.2. NS2

In this second section communications and network configuration must be described. As mentioned before, the file/script must be defined using Tcl programming language. The proposal is the following:

```
# First simulation

# ======================================================
#                 DEFINE OPTIONS
# ======================================================

set val(chan)           Channel/WirelessChannel        ;# channel type
set val(prop)           Propagation/TwoRayGround       ;# radio-propagation model
set val(netif)          Phy/WirelessPhy                ;# network interface type
set val(mac)            Mac/802_11                     ;# MAC type
set val(ifq)            Queue/DropTail/PriQueue        ;# interface queue type
set val(ll)             LL                             ;# link layer type
set val(ant)            Antenna/OmniAntenna            ;# antenna model
set val(x)              210                            ;# X dimension of the topography
set val(y)              210                            ;# Y dimension of the topography
set val(ifqlen)         50                             ;# max packet in ifq
set val(seed)           0.0                            ;# seed
set val(adhocRouting)   AODV                           ;# ad-hoc routing protocol (DSR/AODV)
set val(nn)             10                             ;# number of mobile nodes
set val(nsource)        1                              ;# number of source nodes
set val(coverage)       160                            ;# coverage
set val(sc)             test_trace                     ;# scenario file path
set val(stop)           400.0                          ;# simulation tim
```

Fig. A1-3 NS2 script - DEFINE OPTIONS

The Tcl script is divided in two sections: *DEFINE OPTIONS SECTION* and *MAIN PROGRAM*. General simulation specifications such as simulation time and area or number of nodes are defined in the *DEFINE OPTIONS SECTION* as well as some network and channel features. Moreover the path of the scenario file generated by VanetMobiSim is added in this section too. In the script proposed the path selected is *test_trace* because it is found on the same directory of the script proposed. If not, it would be necessary to consider the corresponding path */directory_1/…/directory_n/test_trace*. Later on, in the *MAIN PROGRAM*, the scenario file will be loaded with the command **source $val(sc)**.

```
# ======================================================
#                   MAIN PROGRAM
# ======================================================

# Create simulator instance
set ns_ [new Simulator]

# Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)          ;# provide the topography object with x and y co-ordinates of the boundary

# Create trace object for ns and nam
set tracefd [open out.tr w]
$ns_ trace-all $tracefd                      ;# write all event in the out.tr file

# Create GOD (General Operations Director)
set god_ [create-god $val(nn)]               ;# GOD objects stores the total number of mobilenodes and a table of shortest number of hops
                                             ;# required to reach from one node to another
```

```
# Global node settings

$ns_ node-config \
-mobileIP                       OFF \
-adhocRouting           $val(adhocRouting) \
-llType                         $val(ll) \
-macType                        $val(mac) \
-ifqType                        $val(ifq) \
-ifqLen                 $val(ifqlen) \
-antType                        $val(ant) \
-channelType            $val(chan) \
-propType                       $val(prop) \
-phyType                        $val(netif) \
-topoInstance                   $topo \
-agentTrace                     ON \
-routerTrace                    ON \
-macTrace                       OFF \
-movementTrace                  OFF \


# Create the nodes and "attach" them to the channel
for { set i 0 } { $i < $val(nn) } { incr i } {
        set node_($i) [$ns_ node]
        $node_($i) random-motion 0                      ;# disable random motion
}

# Define node mobility and traffic model
puts "Loading scenario file..."
source $val(sc)
#puts "Loading connection pattern..."
#source $val(cp)


# Setup traffic flow between nodes
for {set i 0} {$i < $val(nsource)} {incr i} {
        set n [expr {$i + 7}]

        set udp_($i) [new Agent/UDP]
        $ns_ attach-agent $node_($i) $udp_($i)
        set null_($n) [new Agent/Null]
        $ns_ attach-agent $node_($n) $null_($n)

        set cbr_($i) [new Application/Traffic/CBR]
        $cbr_($i) set packetSize_ 512                   ;# in bytes
        $cbr_($i) set interval_ 1.0                     ;# in seconds
        $cbr_($i) set maxpkts_ 10000
        $cbr_($i) set random_ 1                         ;# introduce random "noise" in the scheduled departure times
        $cbr_($i) attach-agent $udp_($i)
        $ns_ connect $udp_($i) $null_($n)

        $ns_ at 0.0 "$cbr_($i) start"
}
# Tell the nodes when the simulation ends
for { set i 0 } { $i < $val(nn) } { incr i } {
        $ns_ at $val(stop).0 "$node_($i) reset"
}

# Simulation end
$ns_ at $val(stop).0002 "stop"
$ns_ at $val(stop).0002 "puts \"end simulation\"; $ns_ halt"

proc stop {} {
        global ns_ tracefd
        $ns_ flush-trace
        close $tracefd
}
$ns_ run
```

Fig. A1-4 NS2 script - MAIN PROGRAM

The *MAIN PROGRAM* includes creation of the simulator instance and trace objects, the nodes settings and the traffic configuration. The proposed network consists on a UDP communication between two nodes: node #0 generates CBR traffic and sent it to node #7. The other eight nodes work as a relay and the mobility model is load from the *test_trace* file, generated by VanetMobiSim.

The performance is the following: the simulation starts and after 400 seconds, NS2 stops the communication and closes the *out.tr* file, which logs all routing events during the simulation.

Once the Tcl script is configured, the next step will be run NS2. To launch the framework, it must invoke the simulator. When simulation ends, the ouput *out.tr* file has been created.

```
roger@roger-laptop:~/NS2$ ns script.tcl
num_nodes is set 10
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
INITIALIZE THE LIST xListHead
Loading scenario file...
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
end simulation
```

Fig. A1-5 NS2 shell launching

A1.3. AWK

In this last section it describes how to filter the *out.tr* output file from NS2 to exctact significant data from the simulation. The proposed awk/filter file is the following:

```
# First awk file

BEGIN {
        # Vars definition
        output="output.txt"
        sent=0
        received=0
        bytes_sent=0
        bytes_received=0
}

{
        # Set variables to the columns of trace file
        action=$1
        node=$3
        prot=$4
        type=$7
        bytes=$8

        # Check all data packets sent by node 0
        if ( action == "s")
        {
                if ( type == "cbr" && prot == "AGT")
                {
                        sent++
                        bytes_sent=bytes_sent+bytes
                }
        }

        # Check all data packets received in node 7
        if (action == "r")
        {
                if (node == "_7_" && prot == "AGT" && type == "cbr")
                {
                        received++
                        bytes_received=bytes_received+bytes
                }
        }
}

END {

        # Formatting the output.

        printf("Transmitted packets: %d;\n", sent) >> output
        printf("Transmitted bytes: %d;\n\n", bytes_sent) >> output

        printf("Received packets: %d;\n", received) >> output
        printf("Received bytes: %d;\n\n", bytes_received) >> output

        printf("Data throughput: %f;\n", received/sent) >> output
        printf("Packet loss: %d;\n\n", sent-received) >> output

        close(output)
}
```

Fig. A1-6 AWK script

With this simple awk file, data about transmitted bytes, received bytes and packet loss will be obtained and saved to *output.txt* file. To launch AWK, it must type

```
roger@roger-laptop:~/NS2$ awk -f filter.awk out.tr
roger@roger-laptop:~/NS2$
```

Fig. A1-7 AWK shell launching

After running AWK, the *output.txt* file contains the following:

Transmitted packets: 397;

Transmitted bytes: 203264;

Received packets: 396;

Received bytes: 210672;

Data throughput: 0.997481;

Packet loss: 1;

ANNEX 2. COMPLETE DESCRIPTION OF NS2 TRACE FILE

NS2 simulator can provide a lot of detailed data of events that occur at the network. If we wish to analyze the data we may generate a trace file which includes all event information occured during network simulation. Then we will be able to handle that trace file and extract relevant information from it aided with scripts or utilities adapted to handle data files, e.g. AWK [6].

A2.1. General description

To know the composition as well as the structure of the trace file is essencial in order to analyze the event information included in. Trace files are organized in rows/lines, where each one is associated to an event. Examples of events are sending or reception of packets and node's movement. Alhtough the structure of each line is slightly different; all of them are composed by some common fields which are presented in the following.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
r	100.00000000	_8_	RTR	---	0	cbr	532	[f2 8 0 800]	-------	[0:0 7:0 30 8]	[0]	1	0

Table A2-1 Event trace format from NS2 trace file

In *Table 10-1* an example of an event trace is presented. The different fields are descripted below

Field	Description
1	*Indicates what type of event occured. We find four possibilites depending on the event:* *s : sent packet* *r : received packet* *f : forwarded packet* *D : dropped packet*
2	*Indicates the simulation time (in second) in which the event occurred*
3	*Indicates in which node the event occurred*
4	*Indicates in which layer the event occured. Some possibilities are:* *MAC : MAC layer* *AGT : Agent (or tranport) layer* *RTR : Routing (or network) layer* *PHY : Phyisical layer*

5	*The reason for dropping a packet if a dropping event occurred. Some possibilities are:* *END: End of simulation* *IFQ : No buffer space in IFQ* *CBK : MAC callback (the link failed)* *NRTE : No route is available*
6	*Indicates the packet global sequence number (packet ID)*
7	*Indicates a descriptive name for the type of packet. Some examples are:* *ack : acknowledgment packet* *udp : UDP packet* *tcp : TCP packet* *cbr : CBR packet*
8	*Indicates the packet size (in bytes)*
9	*Indicates further information about MAC layer. The four subfields are:* *- Time expected to sent packet (in hexadecimal notation)* *- MAC ID of sending node* *- MAC ID of receiving node* *- MAC type*
10	-------
11	*Indicates further information about IP level. The four subfields are:* *- Source IP address and port* *- Destination IP address and port* *- TTL (in hops)* *- Address of the following hop (if needed)*
12	*Indicates the CBR sequence number (CBR packet ID)*
13	*Number of hops for which the packet went through*
14	*Optimum number of hops for which the packet could go through*

Table A2-2 Common fields in NS2 trace file

Considering the common fields descripted in *Table. 10-2*, now we are able to understand the example trace presented in *Table. 10-11*:

A packet is received by the routing layer of the node with *ID 8* at time 100 second. This packet is a *CBR* frame and it has *ID 0* and a length of 532 bytes. In addition, the packet is expected to be sent at 242 seconds (f2 in hexadecimal format), this node has *MAC ID 8*, the sending node has *MAC ID 7* and the MAC type is 800, so IP.

According to the IP fields, the source IP is 0 and the destination IP is 7. Both source and destination ports are 0 and TTL value is 30.

In the following trace format for AODV protocol is analyzed.

A2.2. Traces for AODV routing protocol

In the following sections signaling traces as well as data traces are described.

A2.2.1. Signaling traces: RREQ and RREP

As stated on *Chapter 4*, AODV routing protocols uses special messages during route discovery process: Route Request (*RREQ*) and Route Reply (*RREP*). An example of *RREQ* messages is shown in *Table.10-3*.

1	2	3	4	5	6	7	8	9	10
s	130.00000000	_0_	RTR	---	0	AODV	48	[0 0 0 0]	-------

11	12	13	14	15	16	
[0:255 -1:255 30 0]	[0x2	1	6	[7 0]	[0 14]]	(REQUEST)

1	2	3	4	5	6	7	8	9	10
r	130.09000000	_28_	RTR	---	0	AODV	48	[0 ffffffff 0 800]	-------

11	12	13	14	15	16	
[0:255 -1:255 30 0]	[0x2	1	6	[7 0]	[0 14]]	(REQUEST)

Table A2-3 RREQ messages in AODV

The previous traces show how a *RREQ* message is sent from node with I*D 0* to node with *ID 28*. There are two aspects that differ from trace shown in *Table.10-1*. Firstly, in trace corresponding of the received event, the second value of *field 9* (MAC ID of sending node) is ffffffff. In addition, IP information of source and destination nodes is *0:255* and *-1:255* respectively, as shown in *field 11*. That is because *RREQ* are broadcast messages. And secondly, there are five extra fields (12-16) descripted in *Table.10- 4*.

Field	Description
12	*Indicates type of signaling message. For RREQ is 0x2*
13	*Indicates the number of hops the packet made*
14	*Indicates broadcast message sequent number (broadcast ID)*
15	*Indicates the ID of destination node and broadcast ID from destination node*
16	*Indicates the ID of source node and broadcast ID from source node*

Table A2-4 Added fields for RREQ messages in AODV

Traces for *RREP* messages are quite similar to the previous. There are only two differences.

1	2	3	4	5	6	7	8	9	10
s	131.00000000	_28_	RTR	---	0	AODV	44	[f2 28 7 800]	-------

11	12	13	14	15	
[7:255 0:255 30 0]	[0x4	2	[7 8]	10.000	(REPLY)

1	2	3	4	5	6	7	8	9	10
r	131.09000000	_0_	RTR	---	0	AODV	44	[f2 0 28 800]	-------

11	12	13	14	15
[7:255 0:255 30 0]	[0x4	2	[7 8]	10.000 (REPLY)

Table A2-5 RREP messages in AODV

RREP are not broadcast messages. Hence, *field 9* and *field 11* show now unique destination addresses. In addition, we found extra fields again which are descripted in *Table.10-6*.

Field	Description
12	Indicates type of signaling message. For RREQ is 0x4
13	Indicates the number of hops the packet made
14	Indicates broadcast message sequent number (broadcast ID)
15	Indicates liftetime of the packet

Table A2-6 Added fields for RREP messages in AODV

Finally, it is noticed that the packet ID shown in *field 6* of traces for AODV signaling messages (*RREQ* and *RREP*) is always 0.

A2.2.2. Data traces

Data messages are transmitted using the following procedure as shown in *Table.10-7*.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
s	132.0000	_0_	AGT	---	136	cbr	512	[0 0 0 0]	-------	[0:0 7:0 32 0]	[136]	0	0
r	132.0000	_0_	RTR	---	136	cbr	512	[0 0 0 0]	-------	[0:0 7:0 32 0]	[136]	0	0
s	132.0000	_0_	RTR	---	136	cbr	532	[0 0 0 0]	-------	[0:0 7:0 32 7]	[136]	0	0
r	132.0900	_7_	AGT	---	136	cbr	532	[f2 7 0 800]	-------	[0:0 7:0 30 7]	[136]	1	0

Table A2-7 Data messages for AODV

In the first trace *TCP Agent* of node with *ID 0* sent a packet of 512 bytes to *Route Agent* of the same node. In the second trace *Router Agent* of node with *ID 0* received the packet and in the third trace the packet is sent (adding 20 bytes to its header) to *Route Agent* of destination node with *ID 7*. Finally *TCP Agent* of node with *ID 7* received the packet from *Route Agent* from the same node.

A2.2.3. Packet loss traces: RERR

In *Table.10-8* an example of a dropping event is shown

1	2	3	4	5	6	7	8	9	10	11	12	13	14
s	132.0000	_0_	AGT	---	136	cbr	512	[0 0 0 0]	-------	[0:0 7:0 32 0]	[136]	0	0
r	132.0000	_0_	RTR	---	136	cbr	512	[0 0 0 0]	-------	[0:0 7:0 32 0]	[136]	0	0
s	132.0000	_0_	RTR	---	136	cbr	532	[0 0 0 0]	-------	[0:0 7:0 32 7]	[136]	0	0
D	132.0900	_0_	RTR	IFQ	136	cbr	532	[f2 7 0 800]	-------	[0:0 7:0 30 7]	[136]	0	0

Table A2-8 Drop message for AODV

The procedure is exactly than the previous shown in *Fig. 10-7*. *TCP Agent* of node with *ID 0* sent the data packet to *Route Agent* and enqueue it in the buffer of source node. Then it sent again to *Route Agent* of destination node with *ID 7*. But the packet do not reach destination node because the buffer in source node overflowed. At this point and according to ilustrated on *Chapter 4*, an error message must be sent by the node where the error ocurred, node with *ID 0* in the example. The trace of the error message is shown in *Table.10-9*.

1	2	3	4	5	6	7	8	9	10
s	132.09000000	_0_	RTR	---	0	AODV	32	[0 0 0 0]	-------

11	12	13	14	15	
[0:255 -1:255 1 0]	[0x8	1	[7 0]	0.000000	(ERROR)

Table A2-9 RERR messages in AODV

The *RERR* trace is slighty similar than *RREQ* (*Fig.10- 3*) and *RREP* (*Fig. 10-5*). *Field 11* indicates that the packet sent is a broadcast message and *TTL* is set 1, allowing packet to make only one hop. The extra fields are descripted in *Table.10-10*.

Field	Description
12	Indicates type of signaling message. For RERR is 0x8
13	Indicates the number of hops the packet made
14	Indicates broadcast message sequent number (broadcast ID)
15	Indicates liftetime of the packet (set to 0)

Table A2-10 Added fields for RERR messages in AODV

ANNEX 3. CONFIDENCE INTERVALS

In statistics, a confidence interval (CI) is a particular kind of interval estimate of a population parameter and is used to indicate the reliability of an estimate. It is an observed interval (i.e it is calculated from the observations), in principle different from sample to sample, that frequently includes the parameter of interest, if the experiment is repeated. How frequently the observed interval contains the parameter is determined by the confidence level or confidence coefficient.

A confidence interval with a particular confidence level is intended to give the assurance that, if the statistical model is correct, then taken over all the data that might have been obtained, the procedure for constructing the interval would deliver a confidence interval that included the true value of the parameter the proportion of the time set by the confidence level. More specifically, the meaning of the term "confidence level" is that, if confidence intervals are constructed across many separate data analyses of repeated (and possibly different) experiments, the proportion of such intervals that contain the true value of the parameter will approximately match the confidence level; this is guaranteed by the reasoning underlying the construction of confidence intervals.

A confidence interval does not predict that the true value of the parameter has a particular probability of being in the confidence interval given the data actually obtained. (An interval intended to have such a property, called a credible interval, can be estimated using Bayesian methods; but such methods bring with them their own distinct strengths and weaknesses).

Interval estimates can be contrasted with point estimates. A point estimate is a single value given as the estimate of a population parameter that is of interest, for example the mean of some quantity. An interval estimate specifies instead a range within which the parameter is estimated to lie. Confidence intervals are commonly reported in tables or graphs along with point estimates of the same parameters, to show the reliability of the estimates.

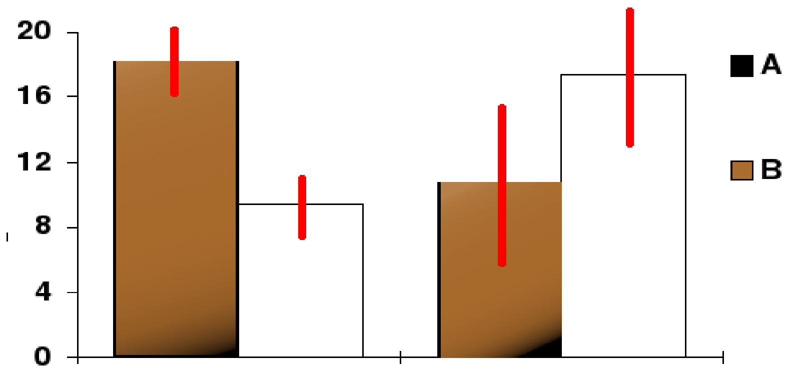


Fig. A3-1 Example of confident interval

For example, a confidence interval can be used to describe how reliable survey results are. In a poll of election voting-intentions, the result might be that 40% of respondents intend to vote for a certain party. A 90% confidence interval for the proportion in the whole population having the same intention on the survey date might be 38% to 42%. From the same data one may calculate a 95% confidence interval, which might in this case be 36% to 44%. A major factor determining the length of a confidence interval is the size of the sample used in the estimation procedure, for example the number of people taking part in a survey.

In the bar chart, the tops ends of the bars indicate observation means and the red line segments represent the confidence intervals surrounding them.

Confidence is defined as 1 - α (1 minus the significance level). Thus, when we construct a 95% confidence interval, we are saying that we are 95% certain that the true population mean is covered by the interval - consequently, of course, we have a 5% chance of being wrong. Any statistic that can be evaluated in a test of significance ("hypothesis testing") can be used in constructing a confidence interval. When constructing a confidence interval two limits, "Upper" and "Lower", are always computed. The upper and lower boundaries for the confidence interval for the t-statistic given above are:

$$\left(\bar{X}-z_{\frac{\alpha}{2}}\cdot\frac{\sigma}{\sqrt{n}},\bar{X}+z_{\frac{\alpha}{2}}\cdot\frac{\sigma}{\sqrt{n}}\right) \tag{A3.1}$$

Where $\bar{X}$ is the mean and σ is the variance of the n samples. Examples $z_{\frac{\alpha}{2}}$ for different % confident interval are:

$$z_{\frac{\alpha}{2}}(90\%) = 1,645$$

$$z_{\frac{\alpha}{2}}(95\%) = 1,96$$

$$z_{\frac{\alpha}{2}}(99\%) = 2,58$$

ANNEX 4. AWK FILTER

```
# First awk file
BEGIN {
       # VARS DEFINITION
       output="salida.xls"

       sig_request0=0
       bytes_request0=0

       sig_reply19=0
       sig_reply0=0

       bytes_reply0=0
       bytes_reply19=0

       sig_error0=0
       bytes_error0=0

       sent=0
       received=0
       bytes_sent=0
       bytes_received=0

       high=0

       route_begin=0
       route_end=0
       route_total=0
       ROUTE=0

       perdidos=0
       bytes_drop=0

       TTL=0
       IFQ=0
       CBK=0
       NRTE=0

       total=0
```

```
	count=1

	saltos=0
	saltos_anterior=0
	paquetes=0
	saltotal=0

	pck1=0
	pck2=0
	pck3=0
	pck4=0
}
{
# set variables to the columns of trace file
	action=$1
	time=$2
	node=$3
	prot=$4
	drop=$5
	id=$6
	type=$7
	bytes=$8
	sigtype1=$25
	sigtype2=$23
	origin=$14
	RTT=$16
# Select all sent packets
if ( action == "s")
{
	if ( type == "AODV")		# Select only signalig sent packets
	{
		if (sigtype1 =="(REQUEST)")
		{
			if (node == "_0_")
			{
				sig_request0++
				bytes_request0=bytes_request0+$bytes
			}
		}
		else if (sigtype2 =="(REPLY)")
		{
			if (node== "_7_")
```

```
                    {
                            sig_reply19++
                            bytes_reply19=bytes_reply19+$bytes
                    }
            }
            else if (sigtype2 =="(ERROR)")
            {
                    # if (node== "_0_")
                    {
                            sig_error0++
                            bytes_error0=bytes_error0+$bytes
                            if (ROUTE == 1)
                            {
                                    route_end=time
                                    ROUTE=0
                            }
                    }
            }
    }
    else if (type == "cbr") # Select data packets sent by node 0
    {
            sent++
            bytes_sent=bytes_sent+bytes
            if (prot == "RTR")
            {
                    start[id]=time
            }
    }
}
# Select all data packets received by destination node (ID 7)
if (action == "r")
{
    if (node == "_7_" && prot == "AGT" && type == "cbr")
    {
            received++
            paquetes++
            bytes_received=bytes_received+bytes
            end[id]=time
            for (k = 0; k < (id-high); k++)
            {
                    end[high+k]=0
            }
```

```
            if (id >= high)
            {
                  high = id+1
            }
            if (time < 700) {pck1++;}
            if (time < 1300){pck2++;}
            if (time < 1900){pck3++;}
            if (time < 2500){pck4++;}
      }
      # Select RREP messages sent by destination node and received by source
node
      if (node == "_0_" && prot == "RTR" && type == "AODV" && sigtype2
=="(REPLY)" && origin == "[7:255")
      {
            sig_reply0++
            bytes_reply0=bytes_reply0+$bytes

            # Miramos si la ruta todavÃa estaba activa
            if (ROUTE == 1)
            {
                  route_end=time
                  ROUTE=0
            }
            # Obsolate route calcs
            saltotal=saltotal+(paquetes*saltos)
            route_total=route_total+(route_end-route_begin)

            # A new route is established
            saltos=(30-RTT)+1
            route_begin=time
            route_end=0
            paquetes=0
            ROUTE=1
      }
}
# Select all drop packets
if ( action == "D" && type == "cbr")
{
      perdidos++
      if (node == "_0_")
      {
            drop_datos++
            bytes_drop = bytes_drop + bytes - 20
```

```
            end[id]=-1
        }
        if (drop == "NRTE") # Can't establish valid route
        {
            NRTE++
        }
        else if (drop == "IFQ") # Queue is full
        {
            IFQ++
        }
        else if (drop == "TTL") # TTL expired
        {
            TTL++
        }
        else if (drop == "CBK") # Current route is no longer available
        {
            CBK++
        }
}
}
END {
# Calculation of e2e time for received data packet
for (n = 0; n < high; n++)
{
        inicio = start[n]
        final = end[n]
        if (inicio < final)
        {
            total = total + (final-inicio)
            count++
        }
}
# Formatting the output.
        printf("%f;\t", received/sent) >> output
        printf("%f;\t", (total*1000)/count) >> output
        printf("%f;\t\t", (saltotal/received)) >> output
        printf("%d;\t", pck1) >> output
        printf("%d;\t", pck2) >> output
        printf("%d;\t", pck3) >> output
        printf("%d;\t\n", pck4) >> output
        close(output)
}
```

ANNEX 5. PROPOSAL ARTICLE FOR JITEL CONGRESS 2012

Performance evaluation of realistic scenarios for Vehicular Ad hoc Networks with VanetMobiSim and NS2

Roger Calzada Cañero, Ahmad Mezher, Carolina Tripp Barba, Mónica Aguilar Igartua

Departament d'Enginyeria Telemática. Universitat Politècnica de Catalunya (UPC)
C/ Jordi Girona 1-3, Mòd. C3, Campus Nord, 08034 Barcelona.
E-mail: {ahmad.mezher, ctripp, maguilar}@entel.upc.edu

***Abstract* - Over recent years, the considerable mobile services sector growth around the world was certainly the major phenomenon in the telecommunications field. Wireless technology has led to the development of new communications systems and multimedia services. Because of the continued growth of the vehicular industry and the increasing demand of road safety a new concept in the communications field was born: vehicular ad hoc networks (VANET). VANETs are receiving more attention from government and car manufacturers due to the wide variety of applications and services they can provide, range from road safety systems to car assistance and Internet acces. However, designing and implementing VANETs is a complex and wide area of research. Thus, since last years R&D community has focused on the study of these networks.**

Keywords- Vehicular Ad Hoc Networks, 802.11p, VanetMobiSim, NS2

1 Introduction

Due to vehicle traffic accidents, the estimated number of deaths is about 1.2 millions people yearly worldwide and of injuries are about forty times of the previous number, without forgetting the traffic congestion that makes a huge waste of time and fuel. With the developments in wireless communications technology, the concept of Vehicular Ad hoc Network (VANETs) has taken the attention all over the world. Such network is expected to be one of the most valuable technology for improving efficiency and safety of the future transportations. Thus, several ongoing research projects supported by industry, governments and academia, have established standards for VANET networks.

In the recent years, many new projects have been proposed trying to evaluate vehicular networks. The ideal evaluation for any network (VANETs could be a choice) related to protocols and applications is to implement it in a real experiment, however this solution has many drawbacks, such as the required expensive investment that could cost implementing such experiment. An alternative evaluation that could achieve to similar results as in the real experiment is the use of simulation tools. A vehicular traffic generator and a network simulator must be coupled in order to generate complete and realistic simulations of VANETs. In this project, we have decided to use VanetMobiSim as a mobility generator, which has a wide variety of driver behavior models and almost realistic road layouts. The network simulator used is NS2, a discrete network simulator with total compatibility with VanetMobiSim.

Basically, this survey is divided in two main parts: Firstly, making a research related to VANET's actual state nowadays in order to find the most appropiate and recommended mobility generator and network simulator reported in the literature. Secondly, due to our previous part, we have decided to use VanetMobiSim [80] as a mobility generator due to its variety models that could be tested and NS2 [63] as a network simulator for being one of the most used by almost all authors and also due to its compatibilty with VanetMobiSim. Using these tools, VanetMobiSim and NS2, we will carry out a performance evaluation of VANETs in several realistic scenarios, giving different values to parameters such as the number of nodes, speed and the propagation model.

2 Current work in VANET

In Europe, several projects are held, joining partners from the industry, governmental agencies and academia. The Car2Car Communication Consortium [CAR04] is a non-profit organization whose mission is to create an open European industry standard for Car2Car communication systems based on wireless LAN components and guarantee European-wide inter-vehicle operability. The NOW (Network on Wheels) project [NOW08] is a German project from the Federal education and research government, which main objective is the implementation of communication protocols and data security algorithms in vehicular network considering the wireless 802.11 technologies and location-based routing in a V2V or vehicle to infrastructure communication context. GST (Global System of Telematics) [GST07] is an EU-funded Integrated Project that is creating an open and standardized end-to-end architecture for automotive telematics services.

3 Urban vehicular mobility model

In this section we describe new mobility models. Each succesive model captures vehicular movement characteristics in increasing levels of detail.

A. Manhattan Model

Manhattan model [BAI04] is a generated-map-based model introduced to simulate an urban environment. Before starting a simulation a map with vertical and horizontal roads are generated. Each road latter includes two lanes, allowing the movement in the two directions (north/south for the vertical roads and east/west for the horizontal ones). At the beginning of a simulation, vehicles are randomly put on the roads. Afterwards, they move continuously according to history-based speeds. When reaching a crossroads, the vehicle randomly chooses a direction to follow. That is, continuing straight forward, turning left or turning right. The probability of each decision is set by the authors respectively to 0.5, 0.25 and 0.25. The security distance is also used in this model and nodes follow the same strategy as in the freeway model to maintain this distance. A vehicle can change a lane at a crossroads. Nonetheless, there is no control mechanism at these points (crossroads), where nodes continue their movements without stopping.

B. City Section Mobility

CSM [DAV00] can be viewed as a hybrid model between Random Waypoint Model (RWP) [JAY08], in which mobile nodes move randomly and freely without restrictions, and Manhattan as it introduces the principle of RWP, especially the pause-time and random selection destination, within a generated-map-based urban area. At each step of the vehicle's movement a random point is selected from the generated road map, towards which it moves following the shortest path. After reaching that destination, it remains there for a pause-time, and then repeats the process. The speed of nodes is constrained by the security distance, along with the maximum speed limit of the road.

C. Stop Sign Model

Contrary to the previous models, SSM [MAH08] integrates a traffic control mechanism. In every crossroads, a stop signal is set, which obliges vehicles to slow down and make a pause there. This model is based on real maps of the TIGER/Lines database, but all roads are assigned a single lane in each direction. A vehicle should never overtake its successor (like in all the models presented before) and should tune its speed to keep the security distance. If many vehicles arrive at an intersection at the same time, they make a queue, and each one waits for its successor to traverse the crossroads. This results in gathering of nodes, and hugely affects the network connectivity as well as the mobility (average speeds). According to the authors, the problem with this model is the unrealistic disposition of the spot signals, since it is impossible to find a region with spot signals at each intersection, therefore, they improved SSM and they proposed TSM.

D. Traffic Sign Model

In TSM model [MAH08], stop signals are replaced by traffic lights. A vehicle stops at crossroads if it encounters a red stoplight; otherwise it continues its movement. When the first vehicle reaches the intersection, the light is randomly turned red with probability p (thus turned green with probability 1-p). If it turns red, then it remains so for a random delay (pause-time) forcing the vehicle to stop, as well as the ones behind it. After the delay, it turns red, and then the nodes traverse the crossroads one after the other until the queue is empty. When the next vehicle arrives at the crossroads the process is repeated.

4 Simulation tools

The following tools take part in the VANET simulation described in the current survey: VanetMobiSim and NS2. Both are open source applications and are descripted in more detail below.

4.1. VanetMobiSim

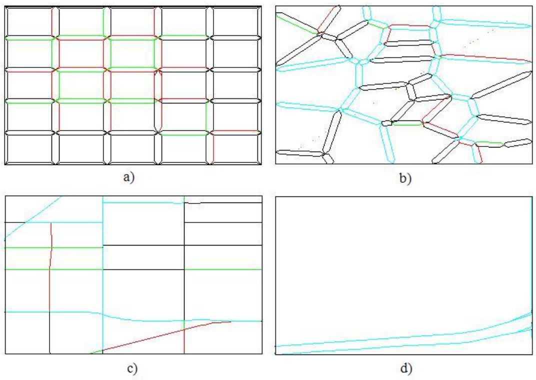
VanetMobiSim [VAN06] is an extension of the CANU Mobility Simulation Environment (CanuMobiSim) [15], which focuses on vehicular mobility, and features realistic automotive motion models at both macroscopic and microscopic levels. At the macroscopic level, VanetMobiSim can import maps from TIGER/Line database, or randomly generate them. VanetMobiSim adds support for multi-lane roads, separate directional flows, differentiated speed constraints and traffic signs at intersections. At the microscopic level, it manages lane changes and vehicle accelerations and decelerations, providing realistic car-to-car and car-to-infrastructure interactions.

4.2. NS2

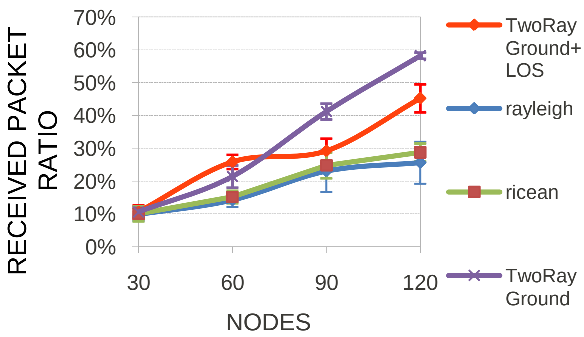
NS-2 [NS10] is an open-source discrete event network simulator that supports both wired and wireless networks, including many MANET routing protocols and an implementation of the IEEE 802.11 MAC layer and it is the most widely used simulator for academic networking research. NS-2 simulates the wireless physical layer and the important parameters that inuence its behavior (e.g., channel fading). The core of NS-2 is written in C++, but users interact with NS-2 by writing TCL scripts, which should contain all of the commands needed to run the simulation. Several of the mobility simulators can generate node descriptions and movement traces suitable for use in NS-2.

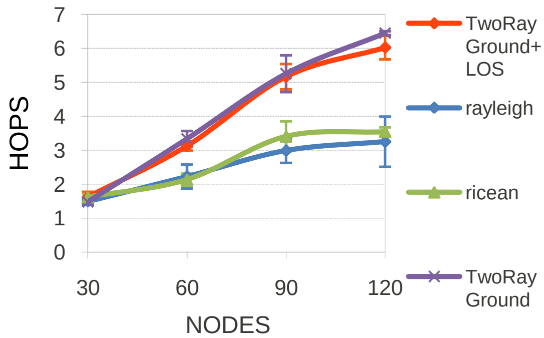
With this couple of simulator, we will carry out an extensive performance evaluation of VANETs in terms of received packet ratio, average number of hops, average end-to-end delay and data throughput, and condering several scenarios, propagation models and MAC protocols.

5 Description of the simulations



6 Results





7 Conclusions

Regarding the four propagation models implemented, we will clasify the results for the four models in terms of data throughput, end-to-end delay and received packet ratio from the best performance to the worst performance as the following: The original Two-Ray ground model, Two-Ray ground with obstacle environment model, Rician fading model and Rayleight model.

References

[BAI04] F. Bai, N. Sadagopan, and A. Helmy, “Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks" *IEEE Jounal on Selected Areas in Communications*, Vol.22, No. 7, Sep 2004

[CAR04] Car2Car Communication Consortium, webpage: http://www.car-to-car.org

[CIC09] Cooperative Intersection Collision Avoidance System (CICAS), webpage: http://www.its.dot.gov/cicas/

[DAV00] V. Davies, "Evaluating mobility models within an ad hoc network", 2000

[GST07] Project GST, http://www.gstforum.org

[INT09] IntelliDrive, official webpage: http://www.intellidriveusa.org/

[IVB05] Integrated Vehicle Based Safety Systems (IVBSS), official webpage: http://www.its.dot.gov/ivbss/

[JAY08] G. Jayakumar, G. Ganapatji, "Reference Point Group Mobility and Random Waypoint Models in Performance Evaluation of MANET Routing Protocols" Protocols" Journals of Computer Systems, Networks, and Communications. Volume 2008 Article ID 860364

[MAH08] A. Mahajan, N. Potnis, K. Gopalan, and A.-I. A. Wang, "Urban mobility models for vanets" In Proc. of 2nd Workshop on Next Generation Wireless Networks (2006)

[NOW08] Network On Weels (NOW), webpage: http://www.network-on-wheels.de/

[NS10] NS-2 simulator http://nsnam.isi.edu/nsnam

[VAN06] VanetMobiSim (Politecnico di Torino) http://vanet.eurecom.fr/

REFERENCES AND BIBLIOGRAPHY

[1] A.Abdi, M.Kaveh, G.Giannakis, « On the Estimation of the K Parameter for the Rice Fading Distribution » IEEE COMMUNICATIONS LETTERS, VOL.5, NO.3 MARCH 2001

[2] K. Abrougui, A. Boukerche, et. al, “Location-Aided Gateway Advertisement and Discovery Protocols for VANETs”.

[3] M. Alam, M. Sher, S. A. Husain, "Integrated Mobility Model (IMM) for VANETs Simulation and Its Impact".

[4] Alshanyour, U. Baroudi, “Random and Realistic Mobility Models Impact on the Performance of Bypass-AODV Routing Protocol”.

[5] D. B. J. Amit Kumar Saha, “Modeling mobility for vehicular ad-hoc networks”

[6] http://www.gnu.org/software/gawk/manual/gawk.html

[7] F. Bai, N. Sadagopan, and A. Helmy, “Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks”

[8] P. Bhagwat, C.E. Perkins, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, Sigcomm (1994)

[9] E. Biagioni, K. Bridges and B.J.S Chee, "PODS: A remote ecological Micro sensor network", (EURASIP Journal on Wireless Communications and Networking archive Volume 2005 Issue 5 - October 2005)

[10] Katrin Bilstrup, Elisabeth Uhlemann, Erik G. Ström, Urban Bilstrup, «Evaluation of the IEEE 802.11p MAC method for Vehicle-to-Vehicle Communication», Centre for Research on Embedded Systems Halmstad University (SWEDEN) , Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th

[11] Bonnmotion Mobility Scenario Generator (Bonn Universität) http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/

[12] R.V. Boppana, P. Konduru, "An Adaptive Distance Vector Routing Algorithm for Mobile Ad Hoc Network", Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and communications Societes, pp.1753-1762, Oct. 2001

[13] BUET – Bangladesh University of Engineering and Technology, http://www.buet.ac.bd/

[14] G. Caizzone, P. Giacomazzi, L. Musumeci, G. Verticale, "A Power Control Algorithm with High Channel Availability for Vehicular Ad-Hoc Networks".

[15] CANU Research Group (Stuttgart University) http://canu.informatik.uni-stuttgart.de/mobisim/

[16] Car2Car Communication Consortium, official webpage: http://www.car-to-car.org

[17] E. Cayirci and T. Coplu, "SENDROM: sensor networks for disaster relief operations management" (Journal Wireless Network Volume 13 Issue 3 - 2007)

[18] U.S. Census Bureau, http://www.census.gov/geo/www/tiger/

[19] W.Chen, S.Cai, "Ad Hoc Peer-to-Peer Network Architecture for Vehicle Safety Communications", IEEE Communications Magazine, pp 100-107, April 2005.

[20] D. R. Choffnes and F. E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks"

[21] CityMob Generador de Patrones de Movimiento (Universidad Valencia) http://www.grc.upv.es/Software/citymob.html

[22] Cooperative Intersection Collision Avoidance System (CICAS), official webpage: http://www.its.dot.gov/cicas/

[23] Coopers (Co-operative System for Intelligent Road Safety), official webpage: http://www.coopers-ip.eu/

[24] CVIS (Cooperative Vehicle-Infrastructure Systems), official webpage: http://www.cvisproject.org/

[25] http://www.dailytech.com/article.aspx?newsid=5257

[26] T. Dang, S. Frolov, N. Bulusu, A. Baptista "Near Optimal Sensor Selection in the COlumbia RIvEr (CORIE) Observation Network for Data Assimilation Using Genetic Algorithms" (DCOSS'07 Proceedings of the 3rd IEEE international conference on Distributed computing in sensor system – Berlin 2007)

[27] V. Davies, "Evaluating mobility models within an ad hoc network"

[28] D. Djenouri, W. Soualhi, E. Nekka, "Mobility models in vehicular ad hoc networks: the overtaking impact".

[29] http://www.dtnrg.org/wiki

[30] M.F. Durate and Y.H. Hu, “Vehicle classification in Distributed Sensor Networks” (Journal of Parallel and Distributed Computing Volume 64 Issue 7 – Orlando July 2004)

[31] http://ec.europa.eu/information_society/activities/esafety/ecall/index_en.htm

[32] http://www.estinet.com/products.php

[33] FleetNet Project – Internet on the road, http://www.et2.tu-harburg.de/fleetnet

[34] FreeSim, 2008. Available at: http://www.freewaysimulator.com/

[35] V. González, A. Los Santos, C. Pinart, F. Milagro «Experimental demonstration of the viability of EEE 802.11b based inter-vehicle communications», Telefónica I+D’s Networked Vehicles Division (SPAIN), TRIDENTCOM 2008 in Belgium – International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities

[36] Project GST, http://www.gstforum.org

[37] Haerri J, Fiore M, Fethi F, Bonnet C. VanetMobiSim: generating realistic mobility patterns for VANETs. Institut Eurécom and Politecnico Di Torino, 2006.

[38] H. Hartenstein, K. Laberteaux, “Vehicular Applications and Inter-Networking Technologies”.

[39] Harvard Sensor Network Lab, http://fiji.eecs.harvard.edu/Volcano

[40] H. Huang, Y. Zhu, X. Li, M. Li, M. Wu, "META: a Mobility Model of MEtropolitan TAxis Extracted from GPS Traces".

[41] IntelliDrive, official webpage: http://www.intellidriveusa.org/

[42] Integrated Vehicle-Based Safety Systems (IVBSS), official webpage: http://www.its.dot.gov/ivbss/

[43] J. Jakubiak, Y. Koucheryavy, "State of the art and research challenges for VANETs", Consumer Communications and Networking Conference, 2008, Jan 10-12. 2008, pp: 912-916.

[44] P. Johanson, T. Larsson, N. Hedman, B. Mielczarek, M. Degermark, “Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks” Proceedings of ACM/IEEE MOBICOM'99, Seattle, WA, August 1999, pp.195-206

[45] D. Johnson, Y. Hu, D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4", RFC4728, February 2007

[46] F. K. Karnadi, Z. H. Mo, and K. chan Lan, “Rapid generation of realistic mobility models for vanet”

[47] Krajzewicz D, Rossel C. Simulation of Urban Mobility (SUMO).

[48] H. Labiod “Wireless Ad Hoc and Sensor Networks” (John Wiley & Sons -2008)

[49] A. Lagar Carvilla “MANET extensions to NS2” Department of Computer Science University of Toronto. Code available at http://lagarcavilla.org/software.html

[50] K. Lan, C. Chou, "Realistic Mobility Models for Vehicular Ad hoc Network (VANET) Simulations".

[51] X. Li, T. Nguyen, R. Martin, “Using Adaptive Range Controls to Maximize 1-Hop Broadcast Coverage in Dense Wireless Networks”.

[52] Liu, et. al, “Assessing the VANET’s Local Information Storage Capability under Different Traffic Mobility”.

[53] J. Luo, et. al, “MI-VANET: A New Mobile Infrastructure Based VANET Architecture for Urban Environment”.

[54] A. Mahajan, N. Potnis, K. Gopalan, and A.-I. A. Wang, “Urban mobility models for vanets”

[55] F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Citymob: a mobility model pattern generator for VANETs". IEEE Vehicular Networks and Applications Workshop (Vehi-Mobi, held with ICC). Beijing, may, 2008

[56] A. Mitra « Lecture Notes on Mobile Communication. A Curriculum Development Cell Project QIP », Department of Electronics and Communication Engineering (Indian Institute of Technology Guwahati) – November 2009

[57] Muñoz C., Aguilar M., “Diseño e implementación de un servidor de vídeo adaptativo en simulador de redes ad-hoc” Proyecto final de carrera, UPC, 2009.

[58] Navarro D., Llàrio X., et. al, “Performance evaluation of vehicular ad-hoc networks over high speed environments using NCTUns” Proyecto final de carrera, UPC, 2010.

[59] NCTUns (National Chiao Tung University, TAIWAN) http://nsl.csie.nctu.edu.tw/nctuns.html

[60] Networking and Emerging Optimization, http://www.neo.lcc.uma.es/

[61] Network On Weels (NOW), official webpage: http://www.network-on-wheels.de/

[62] NS-2.34 source code http://sourceforge.net/projects/nsnam/

[63] NS2 (DARPA Project http://nsnam.isi.edu/nsnam/index.php/User_Information

[64] E.I. Pas, “Recent Advances in Activity-based Travel Demand Modeling” Proceedings of Activity-based Travel Forecasting Conference, June 2-5 1996

[65] C. Perkins, E. Belding-Royer, E. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC3561, July 2003

[66] R.Popescu, I.Radusch, R.A.Rigani “Vehicular-2-X Communications: State-of-the-Art and Research in Mobile Vehicular Ad hoc Networks” (Springer – 2010)

[67] D. Prakash, C. de Morais “Ad Hoc & Sensor Networks” (World Scientific – 2006)

[68] PReVENT, official webpage: http://www.prevent-ip.org/

[69] R.J. Punnoose, P.V. Nikitin, D.D. Stancil, « Efficient Simulation of ricean fading within a packet simulator » September 2000. Code available at http://www.ece.cmu.edu/wireless/downloads/ns2 ricean dist.tgz.

[70] N.N.Qadri, M.Fleury, M.Altaf, M.Ghanbari « Multi-source video streaming in a wireless vehicular ad hoc network » COMSATS Institute of Information Technology, Park Road, Islamabad (Pakistan) – IET Communications 2009 Vol.4, Issue 11, Pages 1300-1311

[71] B. Ramakrishnan, Dr. R. S. Rajesh, R. S. Shaji, «Performance Analysis of 802.11 and 802.11p in Cluster Based Simple Highway Model» (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 1 (5) , 2010, 420-426

[72] SAFESPOT, official webpage: http://www.safespot-eu.org/

[73] P. Sattesh Kumar, S. Ramachandram, "The Performance Evaluation of Genetic Zone Routing Protocol for MANETs", TENCON 2008 – 2008 IEEE Region 10 Conference

[74] L. Schwiebert, S.K.S. Gupta and J. Weinmann, "Research challenges in Wireless Networks of biomedical sensors" (Research challenges in Wireless Networks of biomedical sensors – 2001)

[75] G.Sklyarenko, "AODV Routing Protocol", Seminar Technische Informatik, Institut fur – Freie Universitat Berlin, July 2006

[76] SUMO "Simulation of Urban Mobility. http://sumo.sourceforge.net/"

[77] W. Sun, H. Yamaguchi, and K. Yukimasa, "Gvgrid: A qos routing protocol for vehicular ad hoc networks".

[78] M.Takai, J.Martin, R.Bagrodia « Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks » UCLA Computer Science Department - MobiHoc '01 Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, NY (USA), 2001

[79] J. Tarng, B. Chuang, and F. Wu, "A novel stability-based routing protocol for mobile ad-hoc," IEICE Transactions on Communications, vol. E90-B, no. 4, pp. 876–884, April, 2007.

[80] VanetMobiSim (Politecnico di Torino) http://vanet.eurecom.fr/

[81] VINT (Virtual InterNetwork Testbed) Project http://www.isi.edu/nsnam/vint/

[82] Yi Wang, Akram Ahmed, Bhaskar Krishnamachari, Konstantinos Psounis, «IEEE 802.11p Performance Evaluation and Protocol Enhancement» Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety Columbus, OH, USA. September 22-24, 2008

[83] Shie-Yuan Wang; Chih-Che Lin, "NCTUns 5.0: A Network Simulator for IEEE 802.11(p) and 1609 Wireless Vehicular Network Researches," Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th, pp.1-2, 21-24 Sept.

[84] S.Y. Wang, C. L. Chou, et. Al, "The Protocol Developer Manual for the NCTUns 6.0 Network Simulator and Emulator",

[85] Wei-Yen Lin, Mei-Wen Li, Kun-chan Lan, Chung-Hsien Hsu, «A comparison of 802.11a and 802.11p for V-to-I communication: a measurement study», Computer Science and Information Engineering of Taiwan. 7th International ICST Conference on Heterogeneous

Networking for Quality, Reliability, Security and Robustness (International workshop on DSRC 2010), Houston, Texas, USA November 17-19, 2010

[86] M. Wellens, B.Westphal and P. Mähönen, «Performance Evaluation of IEEE 802.11-based WLANs in Vehicular Scenario», Department of Wireless Networks, RWTH Aachen University. Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th, Page(s):1167 – 1171

[87] http://www.vmware.com/

[88] G.Yan, D.B. Rawat, B.B. Bista, "Provisioning Vehicular Ad hoc Networks with Quality of Service"

[89] F. Zhao, L.J. Guibas "Wireless Sensor Networks" (Morgan Kaufmann – 2004)