Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FINAL DE CARRERA

Performance evaluation of vehicular ad-hoc networks over high speed environments using NCTUns

Estudis: Enginyeria de Telecomunicació

Autors: Daniel Navarro Giménez / Xavier Llàrio Frau

Directora: Mónica Aguilar Igartua

Co-directora: Carolina Tripp Barba

Any: Febrer 2011

Index

Index of figures and tables

Figures

Tables

Glossary of acronyms

Acronym	Definition
ABR	Associativity Based Routing
ACK	ACKnowledgment
ACQUIRE	Active Query Forwarding in Sensor Networks
AFR	Adaptive Face-based Routing
AODV	Ad-Hoc On-Demand Distance Vector
APTEEN	Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network
ASH	Application-aware SWANS with Highway mobility
ASK	Amplitude Shift Keying
A-STAR	Anchor based Street and Traffic Aware Routing
ATT	Advanced Transport Telematics
AU	Application Unit
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
BQ	Broadcast Query
BRP	Based Routing Protocols
BS	Base Station
C2C-C	Car-to-Car Communication
CADR	Constrained Anisotropic Diffusion Routing
CAP	Contention Access Period
CAR	Connectivity Aware Routing
CBR	Constant Bit Rate
CFP	Contention Free Period
CH	Cluster Head
CPU	Central Processor Unit
CSMA/CA	Carrier Sense Multiple Acces With Collission Collission Avoidance

Acronym	Definition
CT	Count Time
CTS	Clear To Send
CVIS	Co-operative Vehicle-Infrastructure Systems
DAB	Digital Audio Broadcast
DAM	Distributed Aggregate Management
DC	Data Collected
DD	Directed Diffusion
DGPS	Differential GPS
DIR	Distance Routing
DSDV	Distance Sequenced Distance Vector
DSP	Digital Signal Processor
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
D-VADD	Direction First Probe VADD
DYMO	Dynamic On demand MANET routing protocol
EBAM	Energy-Based Activity Monitoring
ED	Energy Detection
EDD	Expected Disconnection Degree
EMLAM	Expectation Maximization-Like Activity Monitoring
FCC	Federal Communications Commission
FEC	Forward Error Correction
FOAM	Framework for Open Application Management
FR	Flat Routing
FSR	Fisheye State Routing
FWP	Framework Programme
GAF	Geographic Adaptive Fidelity
GBR	Ground Based Routing
GEAR	Geographic and Energy Aware Routing
GEDIR	Geographic Distance Routing
GeOpps	Opportunistic Geographical Routing
GIS	Geographic Information System
GloMoSim	Global Mobile Information System Simulator
GLONASS	Global Orbiting Navigation Satellite System
GNSS	Global Navigation Satellite System
GOAFR	Greedy Other Adaptive Face Routing
GPCR	Greedy Perimeter Coordinator Routing
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSR	Geographic Source Routing
GST	Global Systems for Telematics
GTS	Guaranteed Time Slot

Acronym	Definition
GUI	Graphical User Interface
HPAR	Hierarchical Power-Aware Routing
HSVN	Hybrid Sensor Vehicular Networks
HT	Hard Threshold
H-VADD	Hybrid Probe VADD
I/O	Input/Output
IARP	Intrazone Routing Protocol
IDM	Intelligent Driver Model
IDSQ	Information-driven sensor querying
IEEE	Institute of Electrical and Electronics Engineers
IERP	Interzone Routing Protocol
IETF	Internet Engineering Task Force
InV	Inter-Vehicle
IP	Internet Protocol
ISO	International Organization for Standardization
ITS	Intelligent Transportation Systems
IVG	Inter-Vehicle Geocast
J-Sim	Java Simulator
LA	Local Aggregator
LAN	Local Area Network
LAR	Location Aided Routing
LDM	Local Dynamic Map
LEACH	Low-Energy Adaptive Clustering Hierarchy
LED	Light Emitting Diode
LML	Local Markov Loop
LQ	Local Query
LQI	Link Quality Indicator
LRWPAN	Low Rate Wireless Personal Area Networks
LS	Link State
L-VADD	Location First Probe VADD
MA	Master Aggregator
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MCFA	Minimum Cost Forwarding Algorithm
MD-VADD	Multi Path Direction First Probe VADD
MECN	Minimum Energy Communication Network
MFR	Most Forward within Radius
MRL	Message Retransmission List
MSDU	MAC Service Data Unit
MURU	Multi-Hop Routing Protocol for Urban VANETs

Acronym	Definition
NCTUns	National Chiao Tung University Network Simulator
NED	NEtwork Description
NS-2	Network Simulator
OBU	On Board Unit
OCDM	Orthogonal Code Division Multiplexing
ODMRP	On Demand Multicast Routing Protocol
OEM	Original Equipment Manufacturer
OFR	Of Face Routing
OMNeT	Objective Modular Network Testbed
OPNET	Optimized Network Evaluation Tool
O-QPSK	Offset Quadrature Phase Shift Keying
PAN	Personal Area Networks
PARSEC	Parallel Simulation Environment for Complex Systems
PBR	Prediction Based Routing
PEGASIS	Power-Efficient Gathering in Sensor Information Systems
PHY	Physical Layer
POMA	Positioning, map services and location referencing
PSSS	Parallel Sequence Spread Spectrum
QoS	Quality of Service
R&D	Research and Development
RAM	Random Access Memory
REQ	Route Errors
RH	Routing Hierarchy
RME	Route Management and Execution
RR	Rumor Routing
RREP	Route Replies
RREQ	Route Requests
RSU	RoadSide Unit
SAR	Sequential Assignment Routing
SDRAM	Synchronous Dynamic Random Access Memory
SEC	Save Energy Consumption
SMECN	Small Minimum Energy Communication Network
SMP	Shared-memory symmetric processor
SNR	Signal to Noise Ratio
SOP	Self Organizing Protocol
SPIN	Sensor Protocols for Information via Negotiation
SPIN-BC	Sensor Protocols for Information via Negotiation – Broadcast Communication
SPIN-PP	Sensor Protocols for Information via Negotiation - Broadcast Communication - Point-to-point
SPIN-EC	Sensor Protocols for Information via Negotiation – Broadcast

Acronym	Definition
	Communication - Energy aware communication
SPIN-RL	Sensor Protocols for Information via Negotiation – Broadcast Communication – with added reliability
SR	Source Routing
SRAM	Static Random Access Memory
ST	Soft Threshold
STRAW	STreetRAndom Waypoint
SUMO	Simulation of Urban MObility
SWANS	Scalable Wireless Ad-hoc Network Simulator
TBF	Trajectory Based Forwarding
TCL	Tool Command Language
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TEEN	Threshold-sensitive Energy Efficient Sensor Network
TI	Texas Instruments
ToA	Time of Arrival
TraNS	Traffic and Network Simulation Environment
TT&C	Tracking, Telemetry & Command
TTDD	Two-Tier Data Dissemination
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
UWB	Ultra Wideband
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VADD	Vehicular Assisted Data Delivery
VANET	Vehicular ad-hoc network
Veins	Vehicles in Network Simulation
VGA	Virtual Grid Architecture
WAN	Wide Area Network
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WRP	Wireless Routing Protocol
WSN	Wireless Sensor Network
WUSB	Wireless Universal Serial Bus
ZRP	Zone Routing Protocol

Abstract

Every year about 1.2 million people die because of traffic accidents [1]. This means that traffic accidents are the fourth cause of mortality in the world. Therefore, several governments and the most important car manufacturers are investing time and money on research and development in order to improve road safety. At this respect, appears the concept of VANET: Vehicular Ad-hoc NETwork.

A VANET is based on smart cars and base-stations that share information via wireless communications. This interchange of data may have a great impact on safety and driving quality but also could be another source of mobile entertainment. This improvement on safety would imply reducing the number of accidents. In addition, the use of wireless communications in mobility would lead to an optimization of transport.

The evolution of VANETs in the last years and their useful applications on the road has been the main reason to develop this project. The great support of many people to this type of wireless networks suggests that VANETs are the networks of the future in mobile environments.

Regarding the project, the first problem encountered is that the network protocol specially designed for VANETs, IEEE 802.11p, is only available in a few of the network simulators and is on phase of development. This fact means that most of the functions are not implemented so it cannot be considered as a mature protocol. As a consequence, a widely used protocol as IEEE 802.11b was chosen and all the tests were performed on NCTUns simulator. So the purpose of this project is to evaluate the performance of VANETs by using 802.11b protocol and AODV routing protocol in a highway scenario. By adjusting different parameters like number of cars, their speed and their range of coverage, variations on measures of loss ratio, throughput and end-to-end delay were detected on the network. Finally, the measures help to know about network communications for each of the cases and their incidence on driving conditions.

Resum

Cada any aproximadament un milió dues-centes mil persones moren en accidents de trànsit [1]. D'aquesta dada es desprèn que els accidents de trànsit són la quarta causa de mortalitat al món. Degut a això, un gran nombre de governs i els majors fabricants de vehicles del món estan invertint temps i diners en recerca i desenvolupament per millorar la seguretat a les carreteres. Amb aquest objectiu, apareix el concepte de VANET: *Vehicular Ad-hoc NETwork*.

Una VANET està basada en vehicles i estacions base intel·ligents que comparteixen informació a través de comunicacions inalàmbriques. Aquest intercanvi de dades podria tenir un gran impacte en la seguretat viària i la qualitat en la conducció però a més a més seria una nova font d'entreteniment mòbil. La millora en seguretat implicaria una reducció en el nombre d'accidents i les comunicacions inalàmbriques usades en mobilitat permetrien una optimització del transport.

L'evolució de les VANETs en els últims anys i les seves aplicacions útils a les carreteres són les principals raons per dur a terme aquest projecte. El gran suport a aquest tipus de xarxes inalàmbriques sembla indicar que les VANETs són les xarxes del futur en entorns mòbils.

En relació al projecte, el primer problema observat és que el protocol que s'usa específicament en VANETs (802.11p) només està disponible en pocs simuladors de xarxa i està en fase de desenvolupament. Per tant, la majoria de les funcions no estan implementades i això fa que el protocol no sigui madur. En conseqüència, es va triar un protocol àmpliament usat com és 802.11b per fer les proves en el simulador NCTUns. L'objectiu del projecte és avaluar el funcionament de VANETs usant el protocol 802.11b i el protocol d'encaminament AODV en un escenari d'autopista. Ajustant diferents paràmetres com el nombre de cotxes, la seva velocitat i el seu rang de cobertura és possible obtenir variacions en les mesures de pèrdues, *throughput* i retard extrem-a-extrem en la xarxa. El resultat final és que les mesures permeten saber quines són les comunicacions que es produeixen a la xarxa per cadascuna de les configuracions i la seva incidència en les condicions de conducció.

:

Resumen

Cada año cerca de un millón doscientas mil personas fallecen en accidentes de tráfico [1]. De este dato se desprende que los accidentes de tráfico son la cuarta causa de mortalidad en el mundo. Debido a esto, un gran número de gobiernos y los mayores fabricantes de vehículos del mundo están invirtiendo tiempo y dinero en investigación y desarrollo para mejorar la seguridad en las carreteras. Con este objetivo, aparece el concepto de VANET: *Vehicular Ad-hoc NETwork*.

Una VANET se basa en vehículos y estaciones base inteligentes que comparten información por medio de comunicaciones inalámbricas. Este intercambio de datos podría tener un gran impacto en la seguridad vial y en la calidad de la conducción pero además sería una nueva fuente de entretenimiento móvil. La mejora en la seguridad implicaría una reducción en el número de accidentes y las comunicaciones inalámbricas utilizadas en movilidad permitirían optimizar el transporte.

La evolución de las VANETs en los últimos años y sus aplicaciones útiles en las carreteras son las principales razones para llevar a cabo este proyecto. El gran apoyo a este tipo de redes inalámbricas parece indicar que las VANETs son las redes del futuro en entornos móviles.

En relación al proyecto, el primer problema observado es que el protocolo específicamente utilizado en VANETs (802.11p) sólo está disponible en pocos simuladores de red y se encuentra en fase de desarrollo. Por lo tanto, la mayoría de funciones no están implementadas y esto hace que el protocolo no sea maduro. En consecuencia, se escogió un protocolo ampliamente utilizado como es 802.11b para realizar las pruebas en el simulador NCTUns. El objetivo del proyecto es evaluar el funcionamiento de VANETs utilizando el protocolo 802.11b y el protocolo de encaminamiento AODV en un escenario de autopista. Ajustando diferentes parámetros como el número de coches, su velocidad y su rango de cobertura es posible obtener variaciones en las medidas de pérdidas, *throughput* y retardo extremo-a-extremo en la red. El resultado final es que las medidas permiten saber cuáles son las comunicaciones que se producen en la red para cada una de las configuraciones y su incidencia en las condiciones de conducción.

1. Introduction and goals

The growing mobility of people and goods incurs in high social costs: traffic congestion, fatalities and injuries. Each year about 1.2 million people die because of traffic accidents [1] around the globe. This statistics place traffic accidents as the fourth cause of mortality in the world. Also, this high number of fatalities and injuries high healthcare costs, more than any other type of injury or disease. Such issues have made traffic safety a major concern to government agencies and automotive manufacturers.

It is in this context that Vehicular Ad-hoc Networks (VANETs) have emerged. These networks are a special kind of Mobile Ad-hoc Networks (MANETs) specialized in vehicular communications. A VANET is based on smart cars and base-stations, which share information via wireless communications. This interchange of data may have a great impact on safety and driving, reducing the number of accidents and helping to optimize transport.

While the original motivation for VANETs was to promote traffic safety, recently it has also become increasingly obvious that VANETs open new vistas for Internet access, distributed gaming, and the fast-growing mobile entertainment industry.

The importance and potential impact of VANETs have been confirmed by the rapid proliferation of consortia involving car manufacturers. This success has been the main motivation to develop this project.

The project consists in seven chapters. Chapter 2 introduces the concept of VANET and describes the different applications, as well as the different projects and standards being developed. Chapter 3 describes the concept of heterogeneous networks. Chapter 4 summarizes the different routing protocols available in VANETs and WSNs. Chapter 5 describes the different simulation tools available for VANET simulation. Chapter 6 includes the procedures of the simulation and the comments on the achieved results. Lastly, chapter 7 contains the conclusions and the future lines of research to follow the study done in this project.

2. Introduction to vehicular networks

2.1 What are vehicular networks?

A vehicular network is a kind of wireless networks that has emerged thanks to advances in wireless technologies and the automotive industry. Vehicular networks are formed between moving vehicles equipped with wireless interfaces that could be of homogeneous or heterogeneous technologies. These networks, also known as VANETs (Vehicular Ad-hoc NETworks), are considered as one of the ad-hoc network real-life applications, enabling communications among nearby vehicles as well as between vehicles and nearby fixed equipment (roadside equipment). Vehicles can be either private, belonging to individuals or private companies, or public means of transport (e.g., buses and public service vehicles such as police cars). Fixed equipment can belong to the government or private network operators or service providers.

The emerging vehicular networks will enable a variety of applications for safety, traffic efficiency, driver assistance and infotainment:

- **Safety:** Vehicular network technologies will be applied to reduce accidents as well as to save lives and reduce injuries.
- **Traffic Efficiency:** Vehicular network technologies will be applied to improve the traffic flow and reduce congestion.
- **Driver Assistance:** Vehicular networks can also provide accurate information and good communications for drivers to improve safety and security.
- **Infotainment:** Multimedia and Internet connectivity facilities for passengers.

2.2 Vehicular network architectures

We can distinguish between three domains: in-vehicle, ad-hoc and infrastructure domain.

The in-vehicle domain refers to a local network inside each vehicle. It is composed of two types of units: an on-board unit (OBU) and one or more application units (AUs).

An OBU is a device with communication capabilities placed in the vehicle. In the other hand, an AU is a device executing a single or a set of applications while making use of the OBU's communication capabilities. The AU and OBU are usually connected with a wired connection, while wireless connection is also possible (using e.g., Bluetooth, WUSB, or UWB).

The ad-hoc domain is a network composed of vehicles equipped with OBUs and roadside units (RSUs) that are stationary placed along the road. OBUs of different vehicles form a mobile ad-hoc network (MANET). OBUs and RSUs can be seen as nodes of an ad-hoc network, respectively, mobile and static nodes. An RSU can be attached to an infrastructure network, which in turn can be connected to the Internet. RSUs can also communicate to each other directly or via multi-hop, and their primary role is the improvement of road safety, by executing special applications and by sending, receiving, or forwarding data in the ad-hoc domain.

Two types of infrastructure domain access exist: RSU and hot spot. Infrastructure, whether roadside or embedded in the highway, is an important part of vehicular network systems because can be used to help provide security and privacy for VANET applications. Fig. 2.1 shows a vehicular network example with different architectures.

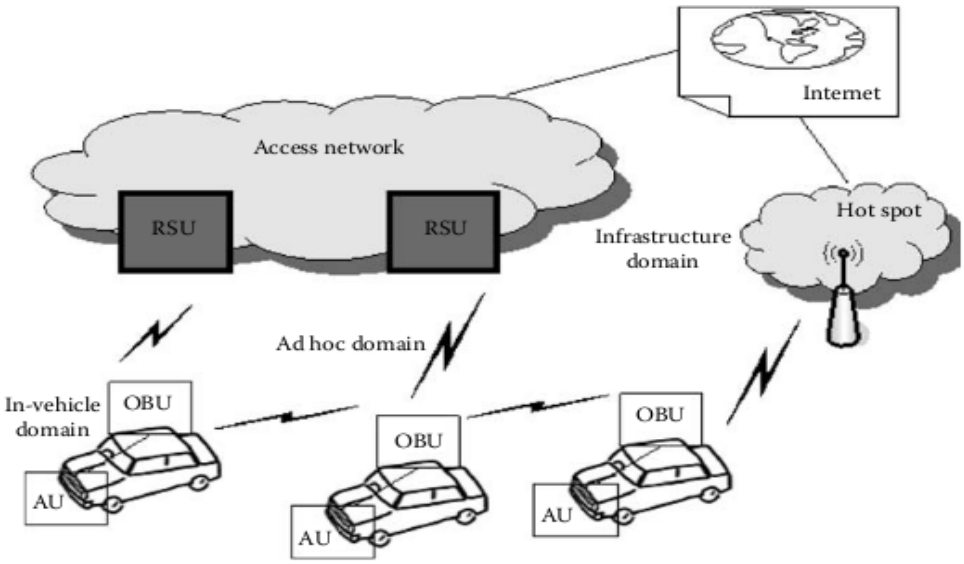


Fig. 2.1: Vehicular network architecture.

2.3 Special characteristics of vehicular networks

Vehicular networks have special behaviour and characteristics that distinguish them from other types of mobile networks:

- **Unlimited transmission power**: The node (vehicle) itself can provide continuous power to computing and communication devices.
- **Higher computational capability:** Operating vehicles can afford significant computing, communication, and sensing capabilities.
- **Predictable mobility:** Vehicles tend to have very predictable movements that are usually limited to roadways. Roadway information is often available from positioning systems and map based technologies such as GPS. Given the average speed, current speed, and road trajectory, the future position of a vehicle can be predicted.

:

- **Potentially large scale:** Vehicular networks can extend over the entire road network and so include many participants.
- **High mobility:** The environment in which vehicular networks operate is extremely dynamic and includes extreme configurations. The density of nodes and their speed can be very high, especially during rush hours.
- **Partitioned network:** Vehicular networks will be frequently partitioned. The dynamic nature of traffic may result in large inter-vehicle gaps in sparsely populated scenarios and therefore in several isolated clusters of nodes.
- **Network topology and connectivity:** Vehicular network scenarios are very different from classic ad-hoc networks. Since vehicles are moving and changing their position constantly, scenarios are highly dynamic.

2.4 Potential Wireless Technologies

With the rapid development of information technologies, there are several candidates that can be potentially used in wireless In-Vehicle (InV), Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications.

Table 2.1 and Table 2.2 summarize the main technologies available for, respectively, InV, and V2V and V2I communications.

Table 2.1: Wireless technologies for InV communications.

	ZigBee	UWB (ultra-wide band)	Bluetooth	Wireless USB (Universal Serial Port)	Wireless CAN
Standard / Technology	Ratified in December 2004	Transmitting information spread over a large bandwidth (>500 MHz)	First launched (1998)	Short-range, high-bandwidth based on the WiMedia Alliance's UWB	CANRF (CAN over RF) / CAN Bridge
Coverage	10 and 75 meters	< 60 cm for a 500 MHz wide pulse, < 23 cm for a 1.3 GHz bandwidth pulse	1 meter, 10 meters, 100 meters	480 Mbps at up to 3 meters and 110 Mbps at up to 10 meters	500 feet (152.4 m)
Bit rate	20-250 kbps per channel	Extremely high data rates 1000+ Mbps	3 Mbps (Version 2.0 + EDR), 53-480 Mbps (WiMedia Alliance (proposed))	480 Mbps at distances up to 3 meters and 110 Mbps at up to 10 meters	20kbps / 52.8 kbps – 164.4 kbps
Applications	Entertainment, smart Lighting control, advanced temperature control, safety & security, sensors, etc	Used at very low energy levels for short-range high-bandwith communications by using a larger portion of the radio spectrum	Connect and exchange information between devices	Game controllers, digital cameras, MP3 players, hard disks and flash drives. Also suitable for transferring parallel video streams	Communication among sensors and ECUs

Table 2.2: Wireless Technologies for V2V & V2I communications.

	GSM/3G	WiFi (Wi-Fi Alliance version of 802.11n)	WiMax	DSRC
Standard / Technology	Third generation cellular technology in 2001	New Wi-Fi technology with MIMO standard in 2009, 802.11n standard in 2009	Broadband technology in 2007	A short to medium range communications
Coverage	kilometers	500 m	5 km	1000m
Bit rate	2-3 Mbps	600 Mbps using MIMO	75 Mbps	6 to 27 Mbps
Applications	Between vehicle and mobile phone communication	Roadside to vehicle and vehicle to vehicle communication	Internet access, Email, VoIP (Voice over IP)	Roadside to vehicle and vehicle to vehicle communication

2.5 Applications of vehicular networks

2.5.1 Safety Applications

Public safety applications are geared primarily toward avoiding accidents and fatalities. Vehicles will be able to communicate with each other, which will allow them to share data that will be helpful to some in-vehicle applications in order to increase safety.

These applications can be classified in five different categories.

2.5.1.1 Intersection Collision Avoidance

The infrastructure has sensors around intersections to collect data about the movement of nearby vehicles. All data collected from sensors is processed and analyzed to determine if there is any unsafe situation that may lead to an accident. If there is risk of an accident, a warning message will be sent to vehicles in the concerned area.

There are many different intersection collision avoidance applications:

- Traffic signal violation warning.
- Stop sign violation warning.
- Pedestrian crossing information warning.
- Left-turn assistant.
- Stop sign movement assistant.
- Intersection collision warning.
- Blind merge warning intersection.

2.5.1.2 Public safety

Minimizing the travel time for emergency teams is the focus of most public safety applications. Other applications in this category focus on requesting help when drivers get into accidents and avoiding potential second accidents.

Some of the most public safety applications are:

- Approaching emergency vehicle warning.
- Emergency vehicle signal pre-emption.
- SOS services.
- Post-crash warning.

2.5.1.3 Sign Extension

In the past few years, new technologies, such as cell phones, have become commonly used while driving. These technologies may distract drivers, which can lead to reckless driving and accidents. Keeping drivers alert while driving is the focus of sign extension applications. Drivers will be alerted about all types of signs on the side of the roads as well as structures, such as bridges in the surrounding area.

There are many different sign extension applications:

- In-vehicle signage.
- Curve speed warning.
- Low parking structure warning.

:

- Wrong way driver warning.
- Low bridge warning.
- Work zone warn.
- In-vehicle AMBER alert.

2.5.1.4 *Information from other vehicles*

Information from other vehicles applications use short-range communications between a host vehicle and other nearby vehicles. This information, for example position heading, is used by in-vehicle applications to complete their functionalities.

There are a lot of different applications that require information from other vehicles:

- Cooperative forward collision warning.
- Vehicle-based road condition warning.
- Emergency electronic brake lights.
- Lane change warning.
- Blind spot warning.
- Highway merge assistant.
- Visibility enhancer.
- Cooperative collision warning.
- Cooperative vehicle–highway automation system.
- Cooperative adaptive cruise control.
- Road condition warning.
- Pre-crash sensing.
- Highway/rail collision warning.
- V2V road feature notification.

2.5.1.5 *Vehicle diagnostics and maintenance*

Vehicle diagnostic and maintenance applications provide alerts and reminders to vehicle owners about safety defects and maintenance schedules for their vehicles.

The two main vehicle diagnostic and maintenance applications are:

- Safety recall notice.
- Just-in-time repair notification.

2.5.2 Comfort applications

The main focus of comfort applications is to make travel more pleasant. This class of applications are motivated by the desire of the passengers to communicate either with other vehicles or with ground-based destinations.

Internet access is the key technology to most comfort applications and, therefore, most work in this category focuses on it. For example, multimedia files, DVDs, music, news, audio books, pre-recorded shows, can be uploaded to the car's entertainment system.

Traveller information applications belong also to this category. For example, the driver could receive local information regarding restaurants, hotels, and the like, when the vehicle approaches a town. Another example is advertisements of gas stations or restaurants along the road.

Nowadays, some vehicle manufacturers provide Internet access in the vehicles via cellular networks. In-vehicle communication with IEEE 802.11 allows all passengers in the vehicle to access the Internet. The main problem with IEEE 802.11 is the limited radio range. It would be very costly to place enough base stations along the roads in order to provide full coverage for IEEE 802.11.

Therefore, a hybrid solution may be feasible, where multi-hop communication can be provided to the closest base station, also called gateway. In this case, the vehicles form an ad-hoc network in order to help each other with the data transfer to and from the gateways.

2.6 Standards and consortiums

The development of vehicular communication systems has been the subject of numerous projects, standardization working groups and industrial consortia around the globe. The projects related in this section have complementary but often similar objectives.

2.6.1 Wireless Access for Vehicular Mobility (WAVE)

Wireless Access for Vehicular Mobility (WAVE) is a set of standards and protocols which goal is to facilitate the provision of wireless access in vehicular environments. It comprehends the IEEE 802.11p, IEEE 1609.1-4 and SAE J2735 standards.

The WAVE standardization process originates from the allocation of the Dedicated Short Range Communications (DSRC) band in the United States. In the 1999, the U.S. Federal Communication Commission reserved 75 MHz in the 5.9 GHz band (5,855 – 5,925) to DSRC to be used exclusively for V2V and V2I communication.

The primary goal was to develop public safety applications that can save lives and improve traffic flow, but private services are also permitted.

2.6.1.1 Assigned spectrum

The DSRC spectrum is structured into seven 10 MHz 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. The rest are service channels (SCH) available for both safety and non-safety usage and a 5MHz guard band.

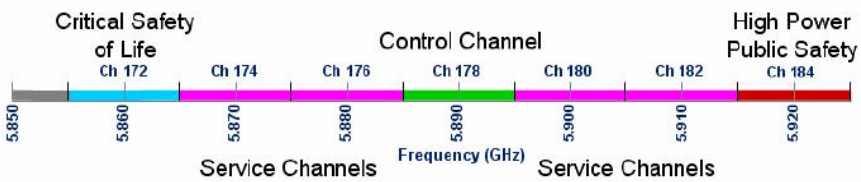


Fig. 2.2: Distribution of DSRC spectrum.

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 10 MHz. These channels are placed in the frequency band between 5,875 and 5,925 GHz. 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. Transmission power in this band is limited to 33 dBm.

:

2.6.1.2 WAVE System Architecture Overview

The WAVE standards cover multiple layers of the open systems interconnection (OSI) model.

- The specification of the physical (PHY) layer and the Media Access Control (MAC) sublayer are addressed in the **IEEE 802.11p**.
- The enhancements provided to the 802.11p MAC to support multichannel coordination are specified on **IEEE 1609.4**.
- **IEEE 1609.3** describes the WAVE services at the network and transport layers.

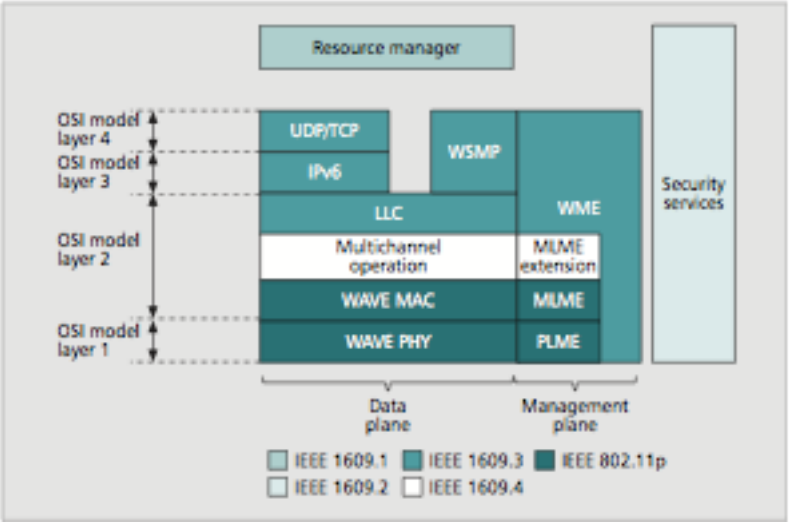


Fig. 2.3: OSI model architecture of WAVE.

The rest of standards have no counterpart in the OSI model:

- **IEEE 1609.1** defines the resource manager.
- **IEEE 1609.2** defines the security services.
- **SAE J2735** defines message sets, data frames and data elements for WAVE.

2.6.1.3 IEEE 802.11p

WAVE physical (PHY) and MAC layers are based on 802.11a. However, some modifications are needed to improve the performance in VANET environments, hence the 802.11p. This new standard is designed to support:

- Longer ranges of operation (up to 1000m).
- High-speed vehicles (up to 500 km/h).
- Extreme multipath environments with many reflections with long delays (up to 5 us).
- Overlapped ad-hoc networks that need to operate with high quality of service.
- Nature of the automotive applications to be supported (e.g. reliable broadcast).

The philosophy of IEEE 802.11p design is to make the minimum necessary changes to IEEE 802.11 PHY so that WAVE devices can communicate effectively among fast moving vehicles in the roadway environment. The physical layer changes basically can be summarized in 3 main changes:

- **10 MHz channel** (half of the 802.11a) in order to have longer guard intervals and therefore, support higher delays. Best performance against multipath errors.

- Although there's a reserved bandwidth in the US for WAVE purposes, there's still a high concern for cross channel interferences. 802.11p includes improvements in the receiver performance requirements in adjacent channel rejections.

- With the same aim, the channel spectrum masks defined in 802.11p are more stringent than the ones demanded of the current IEEE 802.11 radios. There are debates regarding whether and when chipmakers would be able to meet such requirements.

The key parameters of the physical layer of Wi-Fi and IEEE 802.11p are displayed in Table 2.3.

Table 2.3: Comparison between Wi-Fi standards and 802.11p.

	802.11p WAVE	Wi-Fi
Bit rate	3-27 Mb/s	6-54 Mb/s
Communication range (est.)	< 1000 m	< 100 m
Transmission power for mobile (maximum)	760 mW (US) 2 W EIRP (EU)	100 mW
Channel bandwith	10 MHz 20 MHz	1-40 MHz
Allocated spectrum	75 MHz (US) 30 MHz (EU)	50 MHz @ 2.5 GHz 300 MHz @ 5 GHz
Suitability for mobility	High	Low
Frequency band(s)	5.86-5.92 GHz	2.4 GHz, 5.2 GHz
Standards	IEEE, ISO, ETSI	IEEE

2.6.1.4 IEEE 1609.4: Multichannel operation

Every WAVE device must coordinate the multiple service channels. The channel coordination is an enhancement to IEEE 802.11 MAC and interacts with IEEE 802.11 PHY and LLC.

2.6.1.5 IEEE 1609.3: Network services

In the IEEE 1609.3 standard, we find the specification of the functions associated with the LLC, network, and transport layers of the OSI model, and the standard calls them WAVE networking services.

We can functionally divide the WAVE networking services into two sets:

- Data-plane services, whose function is to carry traffic.
- Management-plane services, whose functions are system configuration and maintenance.

:

2.6.1.6 IEEE 1609.2: Security services

In the IEEE 1609.2 standard, we find the security services for the WAVE networking stack and for applications that are intended to run over the stack. These services provide confidentiality, authenticity, integrity and anonymity.

2.6.1.7 IEEE 1609.1: Resource manager

In the IEEE 1609.1 standard, we find the definition of a WAVE application called the resource manager (RM), whose purpose is to give certain processes access to the system communication resources.

The RM is located in either an RSU or an OBU. It receives requests from RMA (Resource Manager Applications) applications that run in computers that are located remotely from its host unit. These RMAs use the resources of one or more OBUs and the RM acts as a broker that relays commands and responses between the RMAs and each OBU.

The RM concept reduces the complexity of the OBUs by freeing them from the requirement of executing applications onboard the vehicle.

2.6.2 IntelliDrive

IntelliDrive, previously known as Vehicle Infrastructure Integration (VII), is a research program funded by the US Department of Transportation (USDOT) with plans to “enhance the safety, mobility and quality of life of all Americans, while helping to reduce the environmental impact of surface transportation".

The IntelliDrive program promotes research and implementation of next generation transportation technologies that could help meet the goals and initiatives stated.

Cooperative Adaptive Cruise Control (CACC) is one of these technologies. CACC is a next generation form of the very familiar cruise control and newly introduced Adaptive Cruise Control (ACC). Another initiative on the IntelliDrive program is to develop vehicle technologies that specifically taken advantage of wireless communication between vehicles and vehicles and the surrounding infrastructure. To achieve better mobility and sustainability, this paper proposed an integration of CACC, IntelliDrive communication, and intelligent traffic signals. An intelligent traffic signal can communicate information with vehicles, such as arrival times, and use arrival information to dynamically adjust the signal timing at an intersection.

2.7 European Projects with VANETs

Since 1980s the European Union is focusing efforts on improving road security, and many projects have been funded with this objective.

The first project that we can consider as the predecessor of VANETs was the PROMETHEUS (Programme for European Traffic with Highest Efficiency and Unprecedented Safety 1987-1995) project with the support of EUREKA (European Research Coordination Agency).

The objective of this first approach to intelligent cars was developing a kind of automatic driving for private cars. Later, it shifted the focus to driver information using in-vehicle systems such as the intelligent co-pilot.

The next big European project was called DRIVE I (Dedicated Road Infrastructure for Vehicle Safety in Europe) (1988-1991) and promoted advanced transport telematics (ATT). The objective of this project was connecting cars with the road-side infrastructure. This project had a continuation (1992-1994) in a project called DRIVE II. These projects supposed a first effort of European society to develop a CAR-2-X communication, but the lack of appropriate technology didn't allow these research groups to find a real solution.

The technological solution that allows us now to focus the research in CAR-2-X communication with the hope to achieve this objective is the development in latest years of wireless technology such as IEEE 802.11.

2.7.1 The Intelligent Car Initiative

The Intelligent Car Initiative is an initiative of the European Union to take profit of latest years radio communications development to improve the quality of life of cars users. The main objective of the initiative is the development of an intelligent vehicle to provide smarter, safer and cleaner roads and cars.

These intelligent cars are supposed to assist drivers in their driving functions, providing value information about the state of roads, weather conditions, traffic jams etc. on real time and in a short-time future helping in the election of the route in an energy efficiency point of view.

This initiative promotes and coordinates the global efforts of the main actors in the development of this future car. Car manufacturers, road operators, telecommunication companies, transport service providers and European regulators. It also helps in the research of this type of technology (in terms of funding and information) and tries to do a social work in order to inform consumers of the future implantation of this type of vehicles.

The European Commission has six future challenges related to vehicular networking. These challenges are:

1. **eCall:** It's an emergency call system that should drastically reduce the time of response of emergency services in an emergency situation. This automatic system will call automatically to the emergency services in case of accident or emergency, with the objective of saving lives thanks to a faster intervention of emergency services. The European Commission is trying to develop this service by the end of this year.
2. **Intersection assistant:** As it's known that many traffic accidents occurs in intersections, and those are a risk point this assistant it's supposed to reduce the number of accidents there. This system will use sensors and CAR-2-X communications to detect cars in the neighbourhoods warning the driver about

:

their presence. These sensors will also communicate with traffic lights to recommend the user a traffic speed that allows him to safe energy.

3. **Wireless local danger warning:** Technology that uses CAR-2-X wireless communication to detect unexpected events on the road.
4. **Lane change assistant and blind spot detection:** System that provides a warning system to the user when he switch the lane of a rail in where another car is detected.
5. **Dynamic traffic management:** Technology that provides real-time information from different sources such as sensors, GPS etc. to manage different strategies to select the best way and avoid traffic jams and unexpected obstacles in the road.
6. **Adaptive cruise control:** Thanks to CAR-2-X communication the vehicle adapts its speed depending of their neighbours' vehicles speed and distance.

2.7.2 Relevant R&D Projects

The European commission developed a policy for sustainable mobility. This policy defines a political framework to ensure a high level of mobility, protection of humans and the environment, technological innovation, and international cooperation.

Based on this policy, the EU is funding projects on this area. Many of these efforts are organized as cooperative projects in the European Framework Programs (FWP), which defines project objectives over time periods of typically 4-7 years.

The major results have been achieved by projects in the 6th FWP such as GST, PReVENT, FleeNet, Internet on the Road and Network on Wheels and in the 7th FWP, such as CIVIS, SAFESPOT, COOPERS.

2.7.2.1 FleetNet – Internet on the Road (2000 – 2003)

Project developed by Daimler and funded by the German Ministry of Education and Research that studied the real possibilities of communication between cars. The project worked in the investigation of different radio technologies IEEE 802.11, UTRA-TDD, and data transmission by radar communications. Finally he used mainly IEEE 802.11 developing a prototype of car-to-car communication platform. Because of the limited communication range of this type of radio communication they investigated multi-hop communication. Due to all difficulties in terms of scalability a position-based was used. In this system each node selects the next hop for a data packet so that the geographical distance to the destination is reduced. These simulations concluded that position-based routing improved topology-based routing protocols (as DSR) in highways and in terms of packet delivery ratio when multi-hop communication was required.

A Linux-based protocol router was developed in the context of this project, using IEEE 802.11a and a planar antenna. The router used GPS trough an Ethernet connection to get GPS information. These real simulations (with real cars and highways) of the project finished with promising results.

2.7.2.2 NoW – Network on Wheels (2004 – 2008)

The successor of FleetNet was born by the same actors with the idea of develop an open communication platform for road safety, traffic efficiency and infotainment applications and analyze strategies to introduce this Car-to-Car communication to European market (with the creation of an European standard). This project tries to unite safety and applications of information and entertainment at the same time, and this adds great difficulties to the project because of different types of requirements of these applications. For this reason NoW distinguished between two types of messages. The ones that gave users periodic information such as beacons or heartbeats, and those that gave users non-periodic information, such as car accidents or unexpected situations in the road.

The network architecture of NoW project was divided in a ad-hoc car-to-car protocol that provided information for road safety applications, and a traditional IP protocol stack for infotainment applications. The network architecture used an IEEE 802.11p radio interface for road safety and an IEEE 802.11a/b/g radio interface for infotainment applications.

This project created a Linux-based communication system prototype of a Car-2-X communication platform.

2.7.2.3 PReVENT – (2004-2008)

PReVENT was an R&D integrated project financed by the European Commission to increase road safety. This project consisted of 13 subprojects. Each one of them investigated different fields to achieve their objective from different sides. Two of this projects included Vehicular Networking, WILLWARN and INTERSAFE.

WILLWARN used different sensors to notify users of unexpected situations on their route. This project generated messages with three kinds of information: Distribution information, event information and position information. To send their messages they used the system developed in the NoW project.

In the INTERSAFE subproject they combined sensor and communication technologies to increase safety at intersections. This project developed an intersection driver warning system.

2.7.2.4 CVIS - (2006 to 2010)

The project called Co-operative Vehicle-Infrastructure Systems (CVIS) is a R&D project financed by the European Commission. The objective of this project was to increase road safety and traffic efficiency giving four new services to users. These systems are COMM (that allows car-2-X communication), POMA (providing positioning systems), COMO (providing monitoring services) and FOAM (linking vehicles with infrastructure).

CVIS considers a wide range of communications technologies such as cellular technologies (GPRS or UMTS), infrared, and wireless technologies. There are not final results for this project, since is not finished yet.

:

2.7.2.5 SAFESPOT – (2006 to 2010)

Another project funded by the European Commission combining a car-to-car and car-to infrastructure communication. This project develops a Safety Assistant that detects potentially dangerous situations in advance and gives information about surrounding environment in space and time.

This project is mainly based on IEEE 802.11 wireless technology and the starting point is the NoW project. The project cooperates with CVIS in the research of a European standard ISO CALM. Another challenge of SAFESPOT is to find a reliable relative positioning system to improve Car-2-X communications. They are considering multiple options such as GPS, Galileo or WLAN.

With the idea of share static and dynamic information SAFESPOT has developed the concept of Local Dynamic Map (LDM). The LDM is a multilayered dynamic representation of the vehicle and everything that surrounds it, collecting information thanks to their sensors.

This project provides users of unexpected events to react safely, and as the previous one, is still ongoing.

2.7.2.6 Carlink (2006 to 2008)

This project developed by the University of Malaga and funded by the Spanish government and the European commission had has main objective the development of an intelligent wireless traffic service platform between cars and supported by wireless transceivers beside the road.

The first objective of this project wasn't the car-to-car communication to avoid unexpected events, but real-time local weather data, urban transport traffic management etc through the wireless communication between cars and local data base station. In order to transmit this kind of information cars could communicate to each other as members of an ad-hoc network.

The radio communications systems being tested on this project were evolutionary extensions of WLANS, WiMAX or mobile communications such as GPRS and the systems were tested on different weather conditions with different topologies.

2.7.2.7 COOPERS - (2006 to 2010)

COOPERS is an integrated R&D project also financed by the European Commission. This project is focused on the traffic management (infrastructure status, traffic jams), using wireless communication between cars, and between cars and infrastructures.

The project is divided in three main parts. The first part that tries to improve the road sensor infrastructure, the second one develops a communication infrastructure in terms of reliability, robustness and real-time capability. The communication systems used are digital audio broadcast (DAB), GPRS, UMTS, WLAN and infrared and microwave technologies. The third part of the project (still not performed) will test the results of this project in the main European highways.

2.7.2.8 SeVeCom - (2006 to 2008)

This European Commission R&D project makes an effort in security and privacy aspects of CAR-2-X communication systems. It made an effort in security architecture for CAR-2-X communication systems. In a later phase, the project will develop solutions for in-vehicle intrusion detection, malfunction detection and data consistency and secure positioning.

2.7.2.9 COMeSafety (2006 to 2009)

This project called Communications for eSafety was born as a support platform of the European Commission to all road-safety stakeholders.

This platform provides coordination and consolidates results from the biggest R&D projects that we have talked before.

2.7.3 European projects in cooperation with car manufacturers

2.7.3.1 Car-2-Car Communication Consortium (www.car-to-car.org)

The Car-to-Car Communication (C2C-C) Consortium is a consortium of major European vehicle manufacturers with the idea of creating a European standard for Car-2-X communication based on wireless communication technology and with the goal of improving road safety and traffic efficiency. This consortium is divided in 6 groups:

WG PHY/MAC (focused on the PHY and MAC protocol layers), WG NET (focused on network and transport protocols), WG ARCH (defines the Architecture protocols for CAR-2-X communication), WG APP (identifies application requirements and protocols), WG SEC (works in the protocols security) and WG STA (groups that works with the standardization issues)

The main car manufacturers that take part of this project are Audi, BMW Group, Daimler, Fiat, Honda, Opel, Renault, Volkswagen and Volvo.

2.7.3.2 eSafetyAware (www.esafetyaware.eu)

ESafetyAware is a consortium that seeks to accelerate the market introduction of life-saving technologies by organizing information campaigns and dedicated events aimed at creating awareness of eSafety benefits among policy-makers and end-users.

In this consortium we find car manufacturers as Hyundai-Kia Motors or Toyota, tire manufacturers as Bridgestone or Continental and governments and User's associations such as the European Commission or the FIA.

Some of the systems that have been developed promoted by eSafetyAware are the Blind Spot Monitoring (that indicates you if there is a car in the lane next to you, and that you are not seeing it), the Speed alert (indicates you if you exceed the speed limit) or the Adaptative Headlights (that optimizes the illumination of the road).

:

3. Heterogeneous networks

3.1 Wireless sensor networks

3.1.1 Introduction to wireless sensor networks

Recent advances in embedded computing systems have led to the emergence of Wireless Sensor Networks (WSN). A WSN is an infrastructure formed by small devices using sensors to monitor physical or environmental conditions, such as temperature, sound or vibration. Each device is in charge of sensing (measuring), computing and transmitting its information. Individually, each node is autonomous and has short range; collectively, they are cooperative and effective over a large area.

A sensor node has four key components. A sensing unit composed by sensors and analog to digital converters, a power unit (usually batteries), a processing and storage unit and a transceiver unit for network connection.

The general functions of each sensor node are the data collection, analysis and transmission, normally using wireless links and multi-hop routing, to a sink point (destination device). The sensor nodes gather the sensor's information, perform some processing and send the data to the clustering node; the clustering nodes receive, carry out further processing and forward the data to the next clustering node (multi hop) or to the base station. The final processing node, also called base station, is in charge of receiving and processing the information sent through the sensor network; hence this latter node must have enough memory, power and computational capacity to correctly perform those functions. This node is usually responsible of connecting the sensor network with other networks, in order to inform an administrator about the data collected. The sensor nodes are organized in clusters to reduce the power consumption, because the energy used to transmit a message from a sensor to the clustering node is lower than the one that would be necessary to transmit the same message from the sensor nodes to the base station in a single hop.

The sensors network has some important limitations due to the small size of nodes. The main constraints are related to power supply issues, computational and memory capacity and the environmental conditions. Batteries are usually the power supply used

by the nodes; therefore the lifetime of the nodes directly depends on the duration of the batteries.

The basic functionalities of the nodes, such as managing the sensors, processing the data collected and transmitting the messages, require power supply. That is why each node normally has two operational modes, slept or idle if it is not needed and awake when it is required. The protocols used to guarantee the communication in the network must take into account these limitations to make the best use of the resources and assure the correct functioning of the network as long as possible.

3.1.2 The IEEE 802.15.4 standard

The IEEE 802.15.4 standard defines the physical layer (PHY) and the Medium Access Control (MAC) sub-layer for low-rate Wireless Personal Area Networks (LRWPAN).

The Wireless Personal Area Networks (WPAN) are intended to transmit information, using wireless links, over short distances. The last version of the standard is the IEEE 802.15.4-2006 developed by the IEEE 802.15 group. An IEEE 802.15.4 LR-WPAN could operate in a star or a peer-to-peer topology formed by full function devices and reduced function devices, with one of the full function devices acting as the PAN coordinator, which can start, finish or route communication through the network. All the devices have unique 64-bit addresses.

3.1.2.1 Physical Layer (PHY)

The PHY sublayer is in charge of the activation and deactivation of the radio transceiver, the energy detection (ED) within the current working channel, the link quality indicator (LQI) for received packets, the channel frequency selection and the data transmission and reception.

The standard defines four PHYs operating in three different frequency bands, the 868 MHz band (868 – 868.6 MHz) for Europe, the 915 MHz band (902 – 928 MHz) available in USA and the 2400 MHz band (2400 – 2483.5 MHz) available worldwide.

The 2400 MHz band provides data rates of up to 250 Kbps using Direct Sequence Spread Spectrum (DSSS) and Offset Quadrature Phase Shift Keying (O-QPSK) modulation.

There are three different modulation techniques specified for the 868/915 MHz bands. The first one employs DSSS and Binary Phase Shift Keying (BPSK) modulation, which permits data rates of up to 20 Kbps in the 868 MHz band and 40 Kbps when operating in the 915 MHz band. The second one is an optional PHY defined to use a multi-code modulation technique called Parallel Sequence Spread Spectrum (PSSS) or Orthogonal Code Division Multiplexing (OCDM) and Amplitude Shift Keying (ASK) modulation; the BPSK modulation is also supported. The data rate of this PHY is up to 250 Kbps for both, the 868 MHz and the 915 MHz bands. The third, also optional, PHY uses DSSS and O-QPSK modulation to achieve data rates of up to 100 Kbps in the 868 MHz band and 250 Kbps in the in the 915 MHz band.

The channel assignment is based in a combination of channel numbers and channel pages; there are 32 channel pages and 27 (0 - 26) channel numbers per channel page. The channel page 0 is defined for the 868/915 MHz bands using BPSK modulation and for the 2400 MHz band using O-QPSK modulation, within this channel page the channel numbers assigned are 0 for the 868 MHz band, from 1 to 10 for the 915 MHz

:

band and 11 to 26 for the 2400 MHz band. From channel pages 1 and 2, 11 channels are used by the 868/915 MHz bands using ASK and O-QPSK modulation respectively, with channels numbered from 1 to 10 for the 915 MHz band and the channel 0 for the 868 MHz band. The rest of channel numbers and channel pages are reserved for future use.

3.1.2.2 MAC sublayer

The MAC sublayer is in charge of the access to the physical radio channel using CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) mechanism, the beacon management, the frame validation, the nodes' association and disassociation. It provides reliable links between two peer MAC entities.

The standard defines four types of frames: data, acknowledgement, beacon and MAC command. The acknowledgment frames are optionally sent to confirm the reception of a data frame or a MAC command frame. The MAC command frames are used to control the network, specifying with the command that an action must be performed within the devices that form the network. The data frames contain the information exchanged within the network. The beacon frames are used for synchronization, to identify the PAN, and to inform about the structure of the superframes. The superframes are delimited by network beacons sent by the coordinator and are divided into 16 slots. The time between the first and the last beacon is the Contention Access Period (CAP); the devices can transmit during this time using a slotted CSMA/CA mechanism. For applications that require specific data bandwidth, the coordinator can assign a period of time called Contention Free Period (CFP) containing Guaranteed Time Slots (GTS) at the end of the CAP.

There are two ways of channel access, with beacons and using superframes or without beacons and using unslotted CSMA/CA channel access mechanism.

3.1.3 Applications

Wireless sensor networks cover a wide range of applications in different areas. Each one has several specific requirements to present a good performance. In this section some applications of WSNs are described.

The applications for Wireless Sensor Networks have no limit in industries and deployments having specific technology requirements such as reliability, battery-life, range, frequencies, topologies, size of the network, sampling rate and sensor use. In annex 8.1 the different types of WSN sensors are described.

3.1.3.1 Transport

Intelligent Transportation Systems (ITS) are a set of sensors installed in different points of a vehicle that provide valuable information about the environment to the user, like weather or road conditions. The ITS can take decision about his path with all this information. Also a sensor can automatically activate the security mechanisms, like a belt or the air bag.

3.1.3.2 Security

In the environment there are some elements that can be dangerous: gas leak, bad electric installation, water or air pollution, etc. The fact that a sensor perceives the

information is an easy fact; the main issue is to perform the communication in a right way with a low cost.

A lot of security systems have shown low efficiency, due to be complex or expensive and to cause problems in their massive implementation, because of the dependency of their battery, not compatible technologies, etc.

3.1.3.3 Domotics

This is one of the most attractive applications of this kind of network. Different devices of different manufacturers could communicate among them; leaving the users free of these typical tasks. The management of the light of the television is an example. Each person can set their own profile that is adaptable to their preferences like temperature, light, computer or music in home or office.

3.1.3.4 Military

One of the areas with most possible applications of general ad-hoc networks, and mainly sensor networks, is the military area. In 2005 a prototype of sensor that can detect snipers was presented. The system was capable to localize the origin of the shot with a good precision.

Also WSN can be used to detect personal mines to and remove or deactivate mines in an enemy camp. In this way, WSN can minimize human risk.

3.1.3.5 Monitorization

The lecture of a counter of water or electricity can be done with a sensor network, with one sensor in the building without a wire connection. One node can gather and send all the information to the companies.

3.1.3.6 Health

Monitorization in real time of vital signs, the sensor is in the body of a patient, so that it checking their vital signs.

At home, a sensor can be a device with buttons that can be use by old people in case of emergency.

3.1.3.7 Environmental

In emergency situations a WSN can measure the water level of a river in areas prone to flooding, earthquakes or volcanic eruptions. In this way, they can send alarm messages to emergency services.

3.2 Hybrid Sensor Vehicular Networks

Thanks to the enormous technological advance in wireless networks, and the extensive attention in VANETs and WSN, it is possible to talk about the combination of these two forms of networks. The goal is that each one can complement the weaknesses of the other. This idea is a big challenge because the characteristics of VANETs and WSN are very different.

:

The nodes in a sensor network are very small, mostly static, restricted in resource and energy but they have good sensing capabilities. On the other hand, the topology in a VANET is very dynamic and they do not have problem with the energy.

The new paradigm is call Hybrid Sensor Vehicular Networks (HSVN) and tries to do a contribution to the next generation network architecture, which deploy sensor nodes within the road environment.

The main idea is to use WSN in the roads; the sensors will be in constant communication among them and will send the information collected to those vehicles driving on the road. The information received by the vehicles could be disseminated around over longer distances. Then the information is back to other sensor network where it is stored for a future use by other vehicles that pass over the place.

3.2.1 Information Flow

3.2.1.1 Information flow within the WSN

The events in the road are captured by the WSN. One node in the WSN is selected like gateway and has to collect the information of all the reporting nodes. The gateway sensor role could change periodically and randomly to save energy consumption.

3.2.1.2 Data transition from WSN to VANET

The gateway sensor is the one that sends the information directly to a mobile node of the VANET. Once a new vehicle is present, data should be transmitted. The sensor has a limited transmission range also a short time to send the data. Hence, all data is transmitted compactly into one or very few frames in order to try to maximize the probability of the transmission will success.

3.2.1.3 Dissemination within the VANET

The most important task in the VANET is to transmit the data that has been obtained the gateway of the WSN. In that way, the gateway of the WSN informs to other vehicles about a possible problem in the path.

Another important purpose is to transport the messages among the VANET and try to send it back to another WSN in a remote place.

3.2.1.4 Information back from VANET to WSN

When the information is distributed in the VANET, it can be sent back from the vehicle to another WSN. This is important, because if a VANET losses connectivity and cannot send the data to other vehicles, the warning messages can be stored in the WSN until other vehicles pass by and retrieve this information again.

3.2.1.5 Data transport by moving vehicles

Besides improving the vehicles interconnection, their spatial movement can be utilized for data dissemination as well. If vehicle density is sparse, the vehicles cache data sampled from the WSN. Based on the importance of the data, the information is injected back to the WSN.

3.2.2 Architecture

The most important issue in a HSVN is to incorporate realistic models of WSN data propagation, traffic flow, and an appropriate VANET message dissemination. Combining those modules is really challenging. In addition, the integration of multiple radio systems like ZIgBee, IEEE 802.15.4 (WSN) and 802.11p (VANET) is also an important objective.

3.2.2.1 Sensor Node Subsystem

In this module there are the detection of the events and the delivery of adequate notifications to a near gateway sensor.

While ordinary sensors report their reading to the near gateway sensor, those gateways are responsible to send that data to a vehicle once it is in their range. All sensors act as gateway at some point to save energy.

The car announces its presence in range using periodic beacon messages. Once such a beacon is received, the collected data is sent to the mobile node immediately. The reason is that a mobile node will be in range for a short period of time.

3.2.2.2 Mobile Node Subsystem

This module is responsible to collect the data from the gateway sensor and to diffuse the information to other vehicles.

The information flows from the sensors to the cars and vice versa, so the communication overhead is reduced.

3.2.3 Conclusions

Wireless sensor networks are more than just a specific form of ad-hoc networks. The stringent miniaturization and cost requirements make economic usage of energy and computational power a significantly bigger issue than in other ad-hoc networks. Moreover, specific applications require a rethinking of some of the basic paradigms with which communication protocols are engineered.

As WSNs are still a young research field, there is much activity still ongoing to solve many open issues. However, at the time of this writing, WSNs are not yet ready for practical deployment, mostly because some of the underlying hardware problems, especially with respect to the energy supply and miniaturization, have no solution yet.

In the other hand, Hybrid Sensor Vehicular Network could be an important tool to warn drivers to have a secure way and to choose routes efficiently in case of problems in the roads or dangerous situations. The direct combination of WSN and VANET generates a more complex infrastructure, but its advantages in road safety will make it an important field of research in the next years.

:

4. Routing protocols

4.1 Routing protocols in VANETs

4.1.1 Introduction

Ad-hoc networks consist of a set of nodes equipped with wireless interfaces, which are able to communicate among them without any kind of network infrastructure. One of the most important features of ad-hoc networks is the concept of wireless multi-hop communications. Unlike traditional wireless networks, mobile nodes are allowed to send messages to destinations that are not within the sender's radio range. When the destination is away, intermediate ad-hoc nodes act as relays to forward data packets to reach their destination. Therefore, nodes need to use a routing protocol to find paths to deliver data messages from sources to destinations. In general, ad-hoc nodes can be mobile, which makes the design of routing protocols a very challenging task. For that reason it is one of the research lines which received most attention over the last years. One classification of wireless networks that covers the concept of ad-hoc multi-hop communications is the following:

1. ***Mobile ad-hoc network (MANET)***: A mobile ad-hoc network (MANET) is a collection of mobile nodes that form on the fly a temporary wireless multi-hop network in a self-organizing way, without relying on any established infrastructure. In MANETs, a pair of nodes exchanges messages either over a direct wireless link, or over a sequence of wireless links including one or more intermediate nodes. Nodes are usually battery operated, making energy efficiency one of the important design issues. Computation and memory resources can also be scarce, so routing protocols for this network should not be too complex in order to save resources. Routing protocols assume that the network is fully connected. That is, if the destination of a data packet lies on a different part of the network, such a packet must be forwarded through relying node to its destination. If that route does not exist, the packet is simply discarded.

2. ***Wireless mesh network (WMN):*** This is a particular case of ad-hoc network in which nodes are like static base stations that are able to communicate using multihop routes. Client devices are mobile and switch among mesh nodes as they move around. Mesh nodes can be equipped with multiple radio interfaces for higher

efficiency. In this case, energy, computation and memory resources are not a big problem. Routing protocols are required to find the best possible route for the aggregated user traffic.

3. ***Wireless sensor network (WSN)***: These networks consists in a large amount of tiny wireless devices that are capable to collect information about the environment, like temperature, humidity, light, movement, etc. The size and transmission capability without a wire allow a fast and flexible display about hundreds of these devices. But also they have the restriction of limited energy, computation power and memory. Therefore, energy efficiency and simple algorithms are factors of paramount importance in these networks. Sensors are usually assumed to have knowledge about their own position and those of their neighbourhood. In many applications, the destination is a sink device that processes the data sensed by the nodes, so that its position is known a priori. When this is not the case, the destination's position is unknown and must be discovered.

4. ***Vehicular ad-hoc networks (VANET)***: A vehicular ad-hoc network (VANET) is a special kind of ad-hoc network where each node in the network is a vehicle properly equipped and its movement follow a road. It can be said that a VANET is one of the emerging applications of MANETs. One of the principal differences between MANETs and VANETs is the speed of the nodes, being faster in VANETs. This causes that the life of the links are shorter in VANETs than in MANETs. In addition, the mobility patterns are constrained by the topology of the roads, streets, speed limits, stops and so on. Cars do not usually have energy constraints, and so they can be equipped with high computing and communication capabilities.

Table 4.1, shows the main differences between these kinds of networks. The table shows that mobility in VANETs is higher and also the limitations of WSN, like computation power, memory capacity and energy.

Table 4.1: Properties of types of Ad-hoc Networks

Property	MANET	WMN	WSN	VANET
Network size	Medium	Moderate	Large	Large
Node's mobility	Random	Static	Mostly static	High, non-random
Energy limitations	High	Very low	Very high	Very low
Node's computation power	-	High	Very low	High
Node's memory capacity	-	High	Very low	High
Location dependency	Low	Very low	High	Very high

As in MANETs, multi-hop communications is one of the most important building blocks of VANETs. The possibility of distributing information very efficiently using neighbouring vehicles becomes a very important feature for many vehicular applications.

Another difference between VANETs and MANETs is related to mobility patterns of the nodes and scalability requirements. Besides, a VANET node has access to very relevant information about its environment, which may be helpful to design routing protocols and take decisions. The main technical limitations for MANET routing solutions in VANET scenarios are:

- **Scalability**. MANETs generally have a limited number of mobile nodes (about one or two hundred). Proactive protocols store routes of all the other nodes in the network within their routing tables. In the case of a VANET, storing routing tables for all vehicles is really impractical.
- **Full connectivity**. This is not realistic in vehicular networks. Although the destination is not reachable at the moment of sending a packet, there could be a path between source and destination. This means that vehicles can move carrying the packet until the destination is eventually reached.
- **Mobility prediction**. Most MANET routing protocols assume random mobility patterns. In the case of VANETs, node movements are restricted by the topology of the streets, speed limits, traffic signals, etc.
- **Anticipation of path breakages**. MANET's use timers in their routing protocols that can adjust and react after a route breakage. In several VANET scenarios, the information about the mobility patterns of neighbouring nodes can help to prevent path breakages before they actually happen.
- **Extensive use of flooding**. Most MANET routing protocols are based on flooding. This operation consumes a lot of bandwidth with control messages and limits very much the performance in large networks such as VANETs. As the number of nodes in a VANET environment is very high, flooding mechanisms must be limited.
- **Nonlocal operation**. In MANETs the creation and maintenance of routing paths usually requires the effort of all nodes in the network. In VANETs, with a potentially large number of nodes, localized routing solutions in which nodes only need information from their neighbourhood are more appealing in terms of scalability, control overhead, and adaptation to different network conditions. However, in order for this to work, the destination of the communication must be known or an efficient location service must be designed.
- **Exploitation of existing knowledge**. Currently VANET nodes are equipped with onboard units providing relevant information about the expected environment. All the information is very important to improve the performance of routing protocols. This is not the case in MANET routing solutions.

4.1.2 Routing protocols in MANETs

Academia and IETF (Internet Engineering Task Force) have developed protocols that deal with routing in such unpredictable networks. Ad-hoc protocols can be classified according to their operation into proactive, reactive, hybrid and geographic protocols.

- Proactive protocols issue periodic messages in order to learn the network topology and create routes to every other node present in the network.
- On demand or reactive routing protocols follow the opposite direction of proactive ones. This approach remains silent until a data flow is about to be sent, and then the process to search a route to destinations starts.
- Hybrid protocol propose to proactively set up routes to the nodes inside a given zone, while letting the process of acquiring routes outside that zone operate on demand.
- Geographic protocols exploit location information to make forwarding decision only based on local information. In order to route a packet, a node must know its own position, the destination's position and the one hop neighbours position. Global position can be found by means of Global Positioning System (GPS). As it will be explained in section 4.3, positions are exchanged between neighbours.

The main features of every type of routing protocol are shown in Table 4.2.

Table 4.2: Comparison between different types of Ad-hoc Routing Protocols.

	Proactive	Reactive	Hybrid	Geographic
Overhead	Very high	Low	Medium	Very low
Connection delay	Very low	Very high	Medium	Very low
Mobility impact	High	High	High	Very high/very low
Scalability (nodes)	Very low	Low	Medium	Very high
Scalability (flows)	Very high	Very low	Medium	Very high

4.1.2.1 Proactive routing protocols

4.1.2.1.1 DSDV (Distance Sequenced Distance Vector)

DSDV is a hop by hop distance vector routing protocol that in each node has a routing table that for all reachable destinations stores the next-hop and number of hops for that destination. As a distance vector based protocol, DSDV requires that each node periodically broadcasts its routing updates.

The advantage with this protocol over traditional distance vector protocols is that DSDV can guarantee loop freedom. To guarantee loop freedom, DSDV uses sequence numbers to tag each route. The sequence number shows the freshness of a route such that routes with higher sequence numbers are favourable. A route R is considered more favourable than R' if R has a greater sequence number or, if the routes have the same sequence number but R has lower hop count. The sequence number is increased when a node A detects that a route to a destination D has broken. So the next time node A advertises its routes, it will advertise the route to D with an infinite hop count and a sequence number that is larger than before.

DSDV basically is distance vector with small adjustment to make it better suited for ad-hoc networks. These adjustments consist of triggered updates that will take care of topology changes in the time between broadcasts. To reduce the amount of information in these packets there are two types of update messages defined: *full* and *incremental dump*. The full dump carries all available routing information and the incremental dump that only carries the information that has changed since the last dump.

4.1.2.1.2 FSR (Fisheye State Routing)

FSR is an implicit hierarchical routing protocol. It uses the "fisheye" technique to reduce routing update overhead in large networks. FSR is functionally similar to LS (Link State) routing in that it maintains a topology map of the entire network. The key difference is the way in which routing information is disseminated. In FSR, link state packets are not flooded. Nodes maintain link state table information received from neighbouring nodes, and periodically exchange it with their local neighbours only (no flooding). Through this exchange process, the table entries with larger sequence numbers replace the ones with smaller sequence numbers. Simulation experiments show that FSR is a simple, efficient and scalable routing solution in a mobile, ad-hoc environment.

4.1.2.1.3 WRP (Wireless Routing Protocol)

WRP is a proactive protocol that introduces the shortest path predecessor node for each destination. This algorithm reduces the number of cases in which a temporary routing loop can occur. For the purpose of routing each node maintains distance table, routing table, link cost table and Message Retransmission List (MRL).

A node can decide to update its routing table after either receiving an update message from a neighbour, or detecting a change in the status of a link to a neighbour. Nodes include in response a list of the update message, and they have to acknowledge the message reception. If there is no routing table change compared with the last update, a hello message has to be sent in order to refresh the connection. At the time of update message reception, the recipient modifies its distance and seeks the best routes based on the received information. MRL list must be updated after each ACK (acknowledgment) reception.

4.1.2.2 Reactive Routing Protocols

4.1.2.2.1 DSR (Dynamic Source Routing)

DSR is a reactive protocol based on the source route approach. Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which to forward the packet. The sender explicitly lists this route in the packet's header, identifying each forwarding "hop" by the address of the next node to which to transmit the packet on its way to the destination host.

When a host needs a route to another host, it dynamically determines one based on cached routing information and on the results of a route discovery protocol. Each mobile host participating in the ad-hoc network maintains a route cache in which it caches source routes that it has learned. If a route is found, the sender uses this route to transmit the packet. If no route is found, the sender may attempt to discover one using the route discovery protocol. While waiting for the route discovery to complete, the host may continue normal processing and may send and receive packets with other hosts. When route maintenance detects a problem with a route in use, route discovery may be used again to discover a new, correct route to the destination. Route Requests (RREQ), Route Replies (RREP), and Route Errors (RERR) are the message types defined by DSR.

When a route to a new destination is needed, the node broadcasts a RREQ to find a route to the destination. A route can be determined when the RREQ reaches either the destination itself. The route is made available by unicasting a RREP back to the origination of the RREQ. Upon the reception of this reply each node in the route updates its routing table. So, a route between the source and destination is built.

4.1.2.2.2 AODV (Ad-hoc On Demand Distance Vector)

AODV algorithm enables dynamic, self-starting, multihop routing between participating mobile nodes wishing to establish and maintain an ad-hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. Route Requests (RREQ), Route Replies (RREP), and Route Errors (RERR) are the message types defined by AODV.

4.1.2.2.3 DYMO (Dynamic On demand MANET routing protocol)

Dynamic On demand MANET routing protocol enables reactive, multihop unicast routing among participating DYMO routers. The basic operations of the DYMO protocol are route discovery and route maintenance.

During route discovery, the source DYMO router initiates dissemination of a RREQ throughout the network to find a route to the destination DYMO router. During this hop-by-hop dissemination process, each intermediate DYMO router records a route to the source. When the destination DYMO router receives the RREQ, it responds with a RREP sent hop-by-hop toward the source. Each intermediate DYMO router that receives the RREP creates a route to the destination, and then the RREP is unicast hop-by-hop toward the source. When the source DYMO router receives the RREP, routes have then been established between the source DYMO router and the destination DYMO router in both directions.

Route maintenance consists of two operations. In order to preserve routes in use, DYMO routers extend route lifetimes upon successfully forwarding a packet. In order to react to changes in the network topology, DYMO routers monitor routers over which traffic is flowing.

When a data packet is received for forwarding and a route for the destination is not known or the route is broken, then the DYMO router of source of the packet is notified. A RERR is sent toward the packet source to indicate the route to that particular destination is invalid or missing. When the source's DYMO router receives the RERR, it deletes the route. If the source's DYMO router later receives a packet for forwarding to the same destination, it will need to perform route discovery again for that destination.

DYMO uses sequence numbers to ensure loop freedom. Sequence numbers enable DYMO routers to determine the temporal order of DYMO route discovery messages, thereby avoiding use of stale routing information.

4.1.2.2.4 ODMRP (On Demand Multicast Routing Protocol)

ODMRP applies "on demand" routing techniques to avoid channel overhead and improve scalability. It uses the concept of "forwarding group", a set of nodes responsible for forwarding multicast data, to build a forwarding mesh for each multicast group. By maintaining and using a mesh instead of a tree, the drawbacks of multicast trees in mobile wireless networks (e.g., intermittent connectivity, traffic concentration, frequent tree reconfiguration, non-shortest path in a shared tree, etc.) are avoided. A soft-state approach is taken to maintain multicast group members. No explicit control message is required to leave the group. The reduction of channel/storage overhead and the relaxed connectivity make ODMRP attractive in mobile wireless networks.

The following properties of ODMRP highlight its advantages: simplicity, low channel and storage overhead, usage of up-to-date shortest routes, reliable construction of routes and forwarding group, robustness to host mobility, maintenance and exploitation of multiple redundant paths, exploitation of the broadcast nature of wireless environments, unicast routing capability, scalability using efficient flooding.

4.1.2.2.5 ABR (Associativity Based Routing)

The Associativity Based Routing protocol is in the family of MANET on demand routing protocols. Its distinct feature is the use of associativity ticks which is required to only form routes based on the stability of nodes, under the fact that there is no use to form a route using a node which will be moving out of the topology and thus making the route to be broken. This protocol adds a new metric known as “associativity degree”. Each node sends periodically a special control message called “Beacon”. When a neighbour node receives this Beacon, it increases its associativity value with respect to the sender. The associativity value of a node becomes null when the node loses its link with the corresponding neighbour. When a node needs a route to a destination, it broadcast a BQ (Broadcast Query). Upon its BQ reception, the appropriate receiver adds its address and its associativity degree to the BQ. The destination node chooses the best route depending on the associativity degrees, and then it sends back a reply to the source. A source’s next hop link failure causes a new BQ REPLY process. When a link fails, due to destination or intermediate nodes mobility, a Local Query packet (LQ [H]) is sent to lunch a partial maintenance request, where H is the hops number till the destination. When de destination receives this packet, it chooses the best partial route, and then sends it to the LQ [H] packet sender.

4.1.2.3 Hybrid Routing Protocols

4.1.2.3.1 ZRP (Zone Routing Protocol)

Zone Routing Protocol is a mixture of a reactive and a proactive protocol. It divides the network into several routing zones and specifies two totally detached protocols that operate inside and between the routing zones: IARP and IERP.

The IntrAzone Routing Protocol (IARP) operates inside a routing zone and learns the minimum distance and routes to all the nodes within that zone. The routing protocol is not defined and it can include any number of proactive protocols, such as distance vector or link-state routing. Different zones may operate with different intrazone protocols as long as the protocols are restricted to those zones. A change in topology means that update information only propagates within the affected routing zones as opposed to affecting the entire network.

The second protocol, the IntErzone Routing Protocol (IERP) it is reactive and is used for finding routes between different routing zones. This is useful if the destination node does not lie within the routing zone. The protocol then broadcasts a RREQ to all border nodes within the routing zone; which in turn forwards the request if the destination node is not found within their routing zone. This procedure is repeated until the requested node is found and a route reply is sent back to the source indicating the route. IERP uses a Bordercast Resolution Protocol (BRP) that is included in ZRP. BRP provides bordercasting services, which do not exist in IP. Bordercasting is the process of sending IP datagram from one node to all its peripheral nodes. BRP keeps track of the peripheral nodes and resolves a border cast address to the individual IP-address of the peripheral nodes. The message that was bordercasted is then encapsulated into a BRP packet and sent to each peripheral node.

4.1.2.4 Geographic Routing Protocols

The work on geographic routing, also called position based routing, started in the late 1980's. The operation of a standard geographic routing algorithm normally comprises the following four phases.

- Determining the destinations coordinates.
- Determining 1 hop neighbours coordinates.
- Determining the next relay.
- Message delivery.

As it can be seen, there are some important assumptions that every geographic routing protocol makes. First, nodes must be able to determine their own position. This can be accomplished by using GPS devices.

4.1.2.4.1 LAR (Location Aided Routing)

Location Aided Routing protocols limit the search for a new route to a smaller "request zone" of the ad-hoc network. This results in a significant reduction in the number of routing messages. The main difference between LAR and DSR is that LAR sends location information in all packets to (hopefully) decrease the overhead of a future route discovery. This protocol uses GPS (Global Positioning System) localization information. Before launching any request, the source node acquires the destination's position information from the GPS, and puts it in the request packet. The broadcast of this packet is limited to the nodes located within the smallest area covering the two nodes; nodes beyond this area will drop this packet if they received it.

4.1.3 Vehicular Ad-hoc Networks (VANETs)

Routing in vehicular ad-hoc networks are special features that must be taken into account when considering routing protocols.

4.1.3.1 Special features of VANET's

Several nodes involved in VANET communications are the vehicles. However, the ad-hoc network is not always isolated and instead it can be attached to an infrastructural deployment. Equipped network interface cards such as IEEE 802.11p can be used to communicate both with neighbouring vehicles and roadside units (RSUs). The onboard unit (OBU) can also be provided with other communication technologies (e.g., 2G/3G or WiMax interfaces), which provide direct access to an operator's network when it is available.

In general, a personal vehicle exhibits the following properties:

a. As we have just seen, a vehicle may have high communication capabilities, depending on the interface cards installed in the OBU.
b. Vehicles are equipped with long-lived batteries, so the energy consumption due to communications is almost negligible.
c. Memory and computational resources are high enough to develop complex algorithms. Here, memory and CPU savings are not a critical concern.
d. Position information may be acquired via geographic positioning systems like GPS.
e. Vehicles can have digital maps of the geographic zone they are travelling around. Moreover, they might be aware of the route that is to be followed.

However, not only particular vehicles are in the network. Public transport systems also can be part of the communication. So, buses could participate in the VANET and additionally provide Internet connection by means of a 3G link with the network infrastructure.

From a routing protocol designer perspective, the availability of infrastructure (either mobile or fixed) allows connectivity to external networks (such as the Internet) and, interestingly, can also be used to route packets inside the VANET.

An important characteristic of vehicular scenarios is that nodes cannot freely move around an area. They have to respect the road layout, traffic signals, and other vehicles movements. Also, the distribution of nodes is not fixed. That is, vehicles usually drive around forming groups, and the distance between groups is sometimes larger than the radio coverage of the VANET wireless interfaces. Also, the traffic pattern can be different depending on the kind of road on which the vehicles are moving, for example:

- **Rural road.** In this environment, the traffic density is low and therefore the resulting ad-hoc network would be highly disconnected. Additionally, the average speed of the vehicles is expected to be moderately low.
- **Urban road.** In this case, there are a moderately high number of vehicles which makes it easier to find a path from source to destination. The vehicles would run at moderately low or high speeds, depending on the specific road.
- **City road.** Traffic density is very high, with many vehicles at very low speeds and with long periods stuck in traffic jams or stopped at traffic signals. The availability of network infrastructure would be very high, with several technological choices to establish communications. Buildings can be an obstacle and cause multipath fading.
- **Highway.** In this case the traffic pattern is clearly different; vehicles are driven at high speeds following a road without crossovers or traffic lights. New automobiles entering the highway must be aware of those already on it.

Traffic patterns can be affected by several causes like populated of the zone, timeframe, day (weekend, holiday, etc), meteorological conditions, accidents or if a street is being repaired. These causes and the market penetration of communication technology in the vehicles (with products available for purchase) must be taken into account by routing protocols and applications designers.

One issue important in the vehicular scenarios is the variety of services that the network can offer. Every service has its own requirements. The more important services are:

- **Safety.** To minimize accidents and number of victims on the road.
- **Advanced driver assistance systems.** Try to prevent possible collisions.
- **Traffic management.** Have available information of the current traffic conditions.
- **Infotainment.** Can use multimedia and entertainment applications.
- **Legacy Internet services.** The persons in the vehicle can connect their devices and use Internet applications.

For all these, the solutions originally designed for MANETs are not appropriate in the currently vehicular environment.

4.1.3.2 Challenges

As it has been seen before, VANETs is a special case of MANET, and the design of a routing protocol is a really challenge. In VANETs, there is so much information available, and it can be used to take decisions. Thus the more specialized the solution is, the better performance of the network would be achieved. An important characteristic of the protocols in MANETs, is that they consider random mobility of the nodes. This is no possible in VANETs, because vehicles move following the roads.

In addition, there are different kinds of scenarios, because the speed is not equal in an urban scenario than in a highway. For that reason, traditional MANET routing protocols have a bad performance in a VANET environment. Another cause of the bad performance of these protocols in a VANET is the use of flooding, because in a VANET there can be many nodes trying to send information, and this method causes a waste of bandwidth and the available bandwidth for services will be reduced.

Several characteristics of reactive, proactive and hybrid protocols make impossible its use in VANETs, in addition that the routes that they create don't support the short period of time that the VANET's route have.

In conclusion, the best type of protocol that has a better performance in VANETs are the geographic routing, where local information is used to create a route. Protocols developed for MANETs and WSNs cannot be used in VANETs, because nodes cannot freely move, and they need to follow the paths allowed by the environment. For all these reasons, routing protocols in VANETs need all information about the road, which helps them to take decisions.

In Table 6 it can be seen a qualitative comparison of the existing VANET routing protocols, classified according with the criteria of objectives, characteristics and assumptions. It can be seen that all the protocols are based in position and required that the vehicles have equipment of position system. Also it can be distinguished the main purpose of each protocol, like the ability to provide QoS.

Table 7: Qualitative comparison on VANET routing protocols.

	Vehicle To Vehicle	Delay Tolent / Sparse	Qos	Internet Connectivity	Position Based / Geographic	Greedy Forwarding	Predictive	Buffering (Carry And Forward)	Street Aware	Traffic Aware (Probabilistic)	Traffic Aware (Real Time)	Node Anchored Routes	Position Anchored Routes	Route Repair Or Recovery	Route Caching	Geographic Marker Messages	Map Required	Positioning System Required	Location Service Required	Transport Routes Required	Transport Schedules Required	Traffic Data Required	Mobile Gateways Required
GSR	✓				✓	✓			✓				✓	✓			✓	✓	✓				
SAR	✓				✓	✓		✓	✓				✓	✓			✓	✓	✓				
A-STAR	✓				✓	✓			✓	✓			✓	✓		✓	✓	✓	✓	✓			
GPCR	✓				✓	✓			✓				✓	✓				✓	✓				
CAR	✓				✓	✓	✓	✓	✓		✓		✓	✓	✓	✓		✓					
VADD		✓			✓	✓	✓	✓	✓	✓							✓	✓	✓			✓	
GeOpps		✓			✓	✓	✓	✓	✓								✓	✓	✓				
MURU	✓		✓		✓		✓	✓	✓					✓	✓		✓	✓	✓				
PBR	✓		✓	✓	✓		✓					✓		✓	✓			✓					✓
	objectives				characteristics												assumptions						

4.1.3.3 GPS

There are many techniques for computing position of mobile nodes. All of these techniques have their pros and cons. But almost every application of this kind of network has the assumption that most of the vehicles in these days have a GPS technology installed.

GPS or Global Positioning System is a Global Navigation Satellite System (GNSS) that allows determining a position of an object, person and vehicles around the world with a precision of meters. This system is currently operated in the Department of Defense in the USA. GPS is composed of 27 satellites (24 are operatives and 3 of backups) operating in orbit around the Earth. Each satellite circles the Earth at a height of 20,200 km. The orbits were defined in such a way that each region of the Earth were covered at least by four satellites in the sky. A GPS receiver is a piece of equipment that is able to receive information constantly sent from the satellites, to estimate its distance to at least four known satellites using a technique called Time of Arrival (ToA), and, finally, to compute its position using trilateration. Once these procedures are executed, the receiver is able to know its latitude, longitude, and altitude.

The main solution for VANET localization is to equip each vehicle with a GPS receiver. This is a reasonable solution because GPS receivers can be installed easily in vehicles, a number of which already come with this technology. However, as VANETs advance into critical areas and become more dependent on localization systems, GPS receivers display some undesirable problems such as not always being available and not being robust enough for some applications.

In order to function properly and compute its position, a GPS receiver needs access to at least three satellite signals for two-dimensional (2D) positioning and at least four satellite signals for a three dimensional (3D) position computation. At first sight, this is not a major issue, because the number of visible satellites usually varies between four and eleven. However, the problem is that these signals are easily disturbed or blocked by obstacles including buildings, rocks, dense foliage, electronic interference, and so on. The result is position inaccuracy or unavailability in dense urban environments (urban canyons), tunnels, indoor parking lots, forests, and any other indoor, underground, or underwater environment.

Also, GPS receivers have a localization error of ±10 to 30 m. While this is a reasonable level of precision for most applications, it is definitely not enough for critical VANET's applications. One positive aspect of these errors is that nearby GPS receivers tend to have the same localization error oriented in the same direction. In other words, nearby GPS receivers have correlated errors. If we put a GPS in an already known location, this GPS receiver can compute its position using the information from the satellites and compare the computed position with its known position. The difference between these two positions can be broadcasted and all nearby GPS receivers can correct their computed positions based on the broadcast differential information. This technique is known as Differential GPS (DGPS), and fixed ground-based reference stations are used to broadcast this differential information. The use of DGPS can lead to a few meters precision, which is sufficient for most VANET's critical applications, but it requires the installation of ground based reference stations in order to work.

Nowadays, there is a global navigation satellite system alternative to GPS, Galileo. It was developed for the European Union and its main objective is to avoid the dependency of GPS and GLONASS. Galileo's system will be of civil use and the full system will consist of 30 satellites and its associated ground infrastructure.

Galileo in addition to servicing of autonomy in navigation and location in space will be interoperable with GPS and GLONASS. The user can calculate its position with a receiver that uses satellites from different constellations. By offering two frequencies as standard, Galileo will provide location in space in real time with a precision of meters, something unprecedented in public systems.

The Galileo global infrastructure will be composed of:

- A constellation of 30 satellites. Each satellite will contain a navigation payload and a search and rescue transponder.
- 30-40 sensor stations.
- 3 control centres.
- 9 Mission Uplink stations.
- 5 TT&C stations.

The range of potential applications for GALILEO is extremely wide. Looking beyond the transport sector, where it will enhance safety, efficiency and comfort, GALILEO's advanced technological features and its commercially oriented services will make it a valuable tool for nearly all economic sectors. Integration with other technologies such as mobile communication or traditional navigation aids will further increase its potential.

Useful applications will benefit both industrialised countries and the developing world. These will include social services to disabled or elderly people, tailored information to people on the move, improved management of all modes of transport, infrastructure and public works management, agricultural and livestock management and tracking, coordination of external staff and even e-banking and e-commerce authentication. But the value of GALILEO is not limited to the economy and companies. GALILEO will also be a key asset for the provision of public services. In addition to the use of the system for features such as rescue operations, crisis management, law enforcement and border control, specific user groups will greatly benefit.

Concrete examples of applications are: guide the blind and people suffering from reduced mobility, monitor children or Alzheimer's sufferers with memory loss, help protect the environment, generalize precision agriculture, guide and rescue explorers, hikers, fishermen or sailing enthusiasts, provide on real-time information on local public transport.

4.1.4 Routing Protocols for Vehicular Ad-hoc Networks

When routing protocols are designed specially focused on VANETs, they can be classified in source routing or geographic routing.

4.1.4.1 Source Routing Based Protocols

4.1.4.1.1 Geographic Source Routing (GSR)

The geographic Source Routing tries to overcome the disadvantages of routing in MANET when applied to VANET in urban scenarios.

The principal idea of GSR protocol is to use the knowledge of the street map of the area where the nodes are moving using a static street map and location information about each node. This information is found in the navigator device of the vehicle. When the topology is known, the protocol can apply Dijkstra's algorithm to discover the

shortest path to the destination. Then the message is send to the closer one hop neighbour.

The message can be lost if there are very few nodes, and so the connectivity is low. This protocol has another problem, as it uses a global initial flooding to determine the location of the destination, thus, the scalability is not guaranteed.

4.1.4.1.2 Spatial Aware Routing (SAR)

SAR is a position based unicast routing protocol that predicts and avoids route recovery caused by permanent networks avoid. SAR relies upon the extraction of a static street map from an external service such as GIS (Geographic Information System) to construct a spatial model for unicast routing.

In SAR, a node determines its location on the spatial model and uses the street information to calculate a shortest path to a packet's destination. When this path is determined, the set of geographic locations to be traveled is embedded into the header of the packet. This protocol introduces a way to avoid losing packets and uses GSR as the basis, but this protocol finds an alternative route to that previously found Dijkstra in the source node. First, it finds the shortest path; then it uses Dijkstra algorithm again after eliminating the edge representing the current street. Thus, another optional path can be found. SAR has a good packet reception radio.

4.1.4.1.3 Anchor based Street and Traffic Aware Routing (A-STAR)

Anchor based Street and Traffic Aware Routing (A-STAR) uses city bus routes as a help to find paths with high probability for delivering. The A-STAR protocol consists of including information about the traffic density of the street edges weights. The main idea is to determine a sequence of streets with a high probability of having enough vehicles to do easy the transmission of the data.

The A-STAR algorithm uses anchor-based unicast routing, which involves inserting a sequence of geographic forwarding points into a packet, through which the packet must travel on its route to the destination. This protocol uses a static street map to route messages around potential radio obstacles. All these information is used to compute an anchor path using Dijkstra's least weight path algorithm.

Packets are routed through alternative paths when routing has problems. Streets with problems are marked as out of service. The packet contains information about the recently discarded street, and uses this information to choice a new route. This information is valid only during some time, since they can be outdated.

4.1.4.1.4 Connectivity Aware Routing (CAR)

The CAR protocol introduces the idea to some problems present in other protocols: The non suitable application of pure geographic routing techniques and the assumption of connectivity between nodes inside the streets that some protocols make when computing the path to follow to the destination.

This protocol has the ability to maintain a cache of successful routes between various sources and destination pairs. Also it predicts positions of destination vehicles, repairs routes as those positions change, and employs geographic marker messages. It uses a preliminary flooding based phase to localize the destination node.

So, a new adaptive beaconing mechanism is introduced to maintain the overhead of control messages independently of the density of the network. Establishing the notion of a guard, a geographic marker message that is buffered and passed from one vehicle to another to proliferate forwarding about a node that has moved to a new location. There are two forms of guards: *standing guard*, which is tied to specific geographic coordinates and *travelling guard* which has initial coordinates, initial time and velocity vector.

The performance of this protocol is not especially good; its main problem is the initial broadcast. However CAR maintains enough states to allow efficient communication between two moving nodes. It provides street awareness, performs basic traffic awareness during path discovery, maintains routes, adapts well in low traffic densities and it does not require map or location services.

4.1.4.2 Geographic Routing Based Protocols

As mentioned before, this kind of protocols is the most used in VANETs. However, they are only used to transmit messages between vehicles in the same street.

These protocols do not use source routing, and the main idea is to use geographic routing directly over the map streets, that helps to take decisions about new directions.

4.1.4.2.1 Greedy Perimeter Coordinator Routing (GPCR)

GPCR eliminates the precondition that assumes that each node knows the complete street map, and does not use flooding. This protocol improves upon GSR by eliminating the requirements of an external static street map.

Seeking to minimize potential radio obstacles, this protocol modifies the typical destination based greedy forwarding strategy such that there are only route messages along streets. In this way, routing decisions are only made at street intersections.

GPCR generates the street map as a graph in which crossroads are considered as a *vertex* and the streets like *edges*. Nodes located in junctions are called *coordinators*.

GPCR solves the problem of determining which nodes are located at intersections defining two heuristic methods that designate those nodes as "*coordinators*". These methods are the neighbour table approach and the correlation coefficient approach.

4.1.4.2.2 Vehicular Assisted Data Delivery (VADD)

VADD is based on the idea of carrying and forwarding. The most important issue is to select a forwarding path with the smallest packet delivery delay. This protocol requires each vehicle to know its own position and also requires an external static street map that includes traffic statistics.

Thus, VADD follows the following basic principles:

- Transmit through wireless channels as much as possible.
- If the packet has to be carried through certain roads, the road with higher speed should be chosen.

Due to the unpredictable nature of vehicular ad-hoc networks, we cannot expect the packet to be successfully routed along the pre-computed optimal path, so dynamic path selection should continuously be executed throughout the packet forwarding process.

Each packet has three modes: Intersection, StraightWay and Destination, based on the location of the packet carrier. The most complex and complicated mode is the Intersection because of the many decisions that are made while in this mode.

There are several variations of VADD, Location First Probe VADD (L-VADD), Direction First Probe VADD (D-VADD), Multi Path Direction First Probe VADD (MD-VADD) and Hybrid Probe VADD (H-VADD).

Simulations carried out in show a better performance of the H-VADD protocols in terms of packet delivery ratio, data packet delay and traffic overhead.

4.1.4.3 Trajectory Based Protocol

Currently, most of the vehicles are equipped with GPS technology to determine its position. Also, with Internet connection, these devices support periodical updates of information such as traffic, street closed, weather, location of restaurants, etc. This information can be useful for a routing protocol.

The information about the trajectory that the vehicle is following is very important for this kind of protocols, because it allows taking proper routing decisions.

4.1.4.3.1 Trajectory Based Forwarding (TBF)

Trajectory Based Forwarding (TBF) is a novel method to forward packets in a dense ad-hoc network that makes it possible to route a packet along a predefined curve. The routing process consists of selecting as next relay the neighbour which is closer to a point in the curve.

There are some aspects to carefully consider. First, there are several different characteristics used to define the best neighbour at each step:

- The neighbour closest to the curve on a straight line.
- The neighbour whose closest point to the curve provides the greatest advance along the curve.
- The node closer to the centroid of candidate neighbours.
- A neighbour randomly chosen between the best three.

Also, when nodes are not stationary, it is possible to know the current directions and speeds of neighbours. In such cases a good approach can be to choose the neighbour whose trajectory fits best with the imaginary curve.

4.1.4.3.2 Opportunistic Geographical Routing (GeOpps)

The GeOpps protocol assumes that the source node already knows the position of the destination by some kind of location database or similar approach. Also it assumes that cars can interchange their expected routes using beacon messages.

Nodes in this protocol follow a predefined trajectory; they use the expected trajectory of neighbours to take routing decisions. The general idea is that if a node finds a neighbour whose trajectory goes closer to the destination's position that its own, it sends the packet to that node.

When a message is being sent from a source node to a destination using GeOpps, intermediate nodes use the following method to select the next hop. Each neighbour vehicle that is following a navigation suggested route calculates its future nearest point to the message destination. It also uses a utility function built into its navigation system to calculate the amount of time required to reach that point. The vehicle that can deliver the packet fastest or closest to the destination will be chosen as the next hop for the message. By choosing nodes whose trajectory goes closer and closer to the destination coordinates, it is expected that data packets will eventually make it to the destination.

The main contribution of this protocol is the exploitation of available information in modern vehicles to efficiently select the next packet carrier. The authors evaluated this approach by using realistic traffic traces generated by a traffic simulator. Their results show good performance in various settings in terms of delivery ratio, delay and overhead with respect to existing algorithms.

4.1.4.4 QoS (Quality of Service)

Currently, routing protocols in VANETs cannot provide a complete QoS. In the strictest sense, a QoS protocol should provide guarantees about the level of performance provided. This is often achieved through resource reservation and sufficient infrastructure. However, in an ad-hoc wireless network, this is a difficult task. With the exception of the potential for roadside units, there is no infrastructure to be relied upon for guaranteed bandwidth.

The dynamic and cooperative nature of an ad-hoc network does not lend itself to resource reservation. Factors such as link delay, node velocity and trajectory, node position, distance between nodes, and reliability of links all contribute to the stability of a particular route. Some performance guarantees can be made in vehicular routing, some algorithms can estimate the duration for which a route will remain connected, and minimize the amount of time required to rebuild the connection if it is broken. With those exceptions, the current suite of QoS VANET's routing protocols is most aptly described as a set of ``best effort'' protocols.

4.1.4.4.1 Prediction Based Routing (PBR)

The Prediction Based Routing (PBR) algorithm is focused on providing Internet connectivity to vehicles by exploring the possibility of mobile gateways with wireless WAN connections that can act as Internet gateways for other vehicles, focusing specifically on highway scenarios. The PBR algorithm assumes that each vehicle has knowledge of its own position through GPS or other means. Just before a route failure is predicted, PBR pre-emptively seeks a new route to avoid loss of service.

To communicate to a location on the Internet, a node checks its routing table for an existing route. Like many reactive MANET protocols, if the node finds no existing route, the node broadcasts a RREQ message with a limited number of hops. If multiple gateways are found, the source node chooses the route with the shortest number of hops. Once the route to the gateway is established, communication begins.

Regardless of the effectiveness of the PBR algorithm in the mobile gateway situation, it is unknown how realistic the situation itself is. It is unclear how a vehicle would be motivated to share its wireless Internet connection with others, when that connection is likely to be costly. In addition, the bandwidths of the mobile gateway's wireless WAN connections would need to be significant enough to support the bandwidth demand of numerous client vehicles.

4.1.4.4.2 Multi Hop Routing Protocol for Urban VANETs (MURU)

The Multi-Hop Routing Protocol for Urban VANETs (MURU) balances hop minimization with the ability to provide a robust route connection. In doing so, a new metric called the `expected disconnection degree' (EDD), is introduced to estimate the quality of a route based on factors such as vehicle position, speed, and trajectory. MURU requires each vehicle to know its own position and to have an external static street map available. The presence of an efficient location service is also assumed.

EDD is an estimation of the probability that a given route might break during a given time period. Using this measure, a low EDD is preferred. Intuitively, nodes moving in similar directions at similar speeds are more likely to maintain a stable route. Given certain assumptions about vehicle traffic, routes with very small and very large routes have higher packet error rates. The formula for calculating EDD takes these factors into account to make a prediction about the breakability of a route.

To find a route to a destination, a source node calculates the shortest trajectory to the destination, based on their locations and the static street map. It then initiates a RREQ message, broadcasting it in a rectangular `broadcast area' that encloses the shortest trajectory and is bounded by the positions of the source and destination. Nodes outside of the `broadcast area' will drop the packet.

If every node receiving the RREQ message immediately re-broadcasts it, the message overhead would quickly become not scalable. A node receiving the RREQ will wait for a calculated backoff delay that is directly proportional to the EDD between the previous forwarder of the RREQ and the current node. During this backoff delay, the node listens for RREQ messages at other nodes. If during that window of time, the node overhears a counterpart to this RREQ whose EDD is smaller than its own EDD to the source, then it will drop the RREQ. When the destination finally receives some RREQ messages from different routes, it selects the route with the smallest EDD.

MURU is loop-free and that MURU always chooses a path from source to destination with the smallest EDD, aims to provide a quality route that delivers a high percentage of packets while controlling overhead and delay.

4.1.5 Conclusions

Along this chapter, we have seen some of the principal routing protocols in MANETs, showing their main characteristics and their classification. Also we have seen that these protocols do not present a real solution if used in VANETs.

Because VANETs present several particular properties, they need routing protocols able to solve their specific problems. A VANET should prevent the use of flooding because scenarios include a big number of nodes. A VANET cannot guarantee a full connectivity, so when we design a new protocol we must take into account that a path can only exist during a short time due the high mobility of nodes. Taking into account

information like car position, trajectories, current heading and speed is very important. A solution works better using this existing information.

With the incremented number of vehicles on the streets today, the potential for vehicular ad-hoc networks is enormous. Due to the great interests in this area (e.g. safe driving, better route path planning), this technology has a promise bright. Routing protocols for VANETs have separated themselves from other mobile routing protocols, due to the main characteristics of communication on roadways. Many distinguishing qualities of this environment are not yet explored, leaving opportunities for further research in the area.

4.2 Routing protocols in WSN

4.2.1 Classification

Based on the network structure

- Flat-based routing: All nodes have equal roles or functionality.
- Hierarchical-based routing: Nodes play different roles in the network.
- Location-based routing: Sensor positions are exploited to route data in the network.

Depending on the protocol operation

- Multipath-based: These routing protocols use multiple paths rather than a single path in order to enhance the network performance.
- Query-based: In this kind of routing, the destination nodes propagate a query for data through the network and the node which has the data requested sends it to the node which initiates the query.
- Negotiation-based: The main objective of these protocols is to suppress duplicate information and avoid sending redundant data by using negotiation messages.
- QoS- based: In this kind of protocols, a balance between energy consumption and data quality is applied.
- Coherent-based: These protocols deal with the data processing component.

Depending on how the source finds a route to the destination

- Proactive: Routes are computed before they are really needed.
- Reactive: Routes are computed on demand.
- Hybrid: Combination of proactive and reactive.

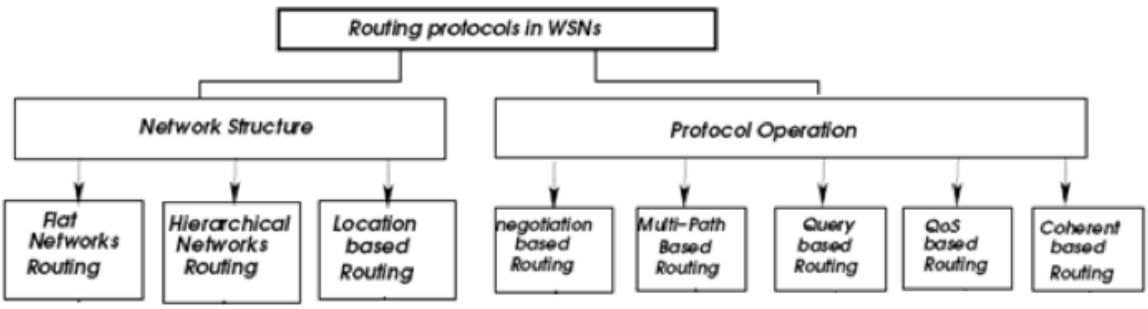


Fig. 4.1: Classification of WSN routing protocols.

4.2.2 Network structure based protocols

4.2.2.1 Flat routing

The basic premise of flat routing is that each node plays the same role and sensor nodes collaborate together to perform the sensing task. Due to the large number of such nodes, it is not feasible to assign a global identifier to each node. Therefore, data centric routing is used, where the base station sends queries to certain regions and waits for data from the sensors located in these regions.

4.2.2.1.1 Sensor Protocols for Information via Negotiation (SPIN)

The information at each node is disseminated to every node in the network, assuming that all nodes are potential base-stations.

The SPIN [49] family of protocols uses data negotiation and resource-adaptive algorithms. Nodes assign a high-level name to completely describe their collected data (called meta-data) and perform meta-data negotiations before any data is transmitted. This assures that there is no redundant data sent throughout the network.

In addition, SPIN has access to the current energy level of the node and adapts the protocol it is running based on how much energy is remaining.

Main ideas:

- Negotiation: Sending data that describe the sensor, not all the data. Distribute data that other nodes do not have.
- Resource adaptation
- Three types of messages:
    - ADV, to advertise new data
    - REQ, to request data
    - DATA, the actual message

Other protocols of the SPIN family:

- SPIN-BC
- SPIN-PP
- SPIN-EC
- SPIN-RL

4.2.2.1.2 Directed Diffusion (DD)

Directed Diffusion [49] is a data-centric (DC) and application-aware paradigm, all data generated by sensor nodes is named by attribute-value pairs.

DC paradigm: It combines the data coming from different sources (in-network aggregation) by eliminating redundancy, minimizing the number of transmissions; thus saving network energy and prolonging its lifetime.

Sensors measure events and create gradients (attribute value and direction) of information in their respective neighbourhoods. The base station requests data by broadcasting interests. Interest describes a task required to be done by the network. The interest is broadcasted by each node to its neighbours, and these reply setting up

a gradient towards the senders. This process continues until gradients are setup from the sources back to the BS.

The strength of the gradient may be different towards different neighbours resulting in different amounts of information flow.

Phases of directed diffusion:

- Sending interests.
- Building gradients.
- Data dissemination.

The differences with SPIN are:

- BS sends queries to the sensor nodes by flooding some tasks. In SPIN, however, sensors advertise the availability of data allowing interested nodes to query that data.
- All communication in directed diffusion is neighbour-to-neighbour with each node having the capability of performing data aggregation and caching. Unlike SPIN, there is no need to maintain global network topology. However, directed diffusion may not be applied to applications (e.g., environmental monitoring) that require continuous data delivery to the BS.

4.2.2.1.3 Rumour routing

Rumour routing [49] is a variation of directed diffusion and is mainly intended for applications where geographic routing is not feasible.

The key idea is to route the queries to the nodes that have observed a particular event rather than flooding the entire network to retrieve information about the occurring events.

When a node detects an event, it adds such event to its local table, called events table, and generates an agent. Agents travel the network in order to propagate information about local events to distant nodes. When a node generates a query for an event, the nodes that know the route, reply to the query by inspecting its event table.

One advantage of rumour routing is that there is no need to flood the whole network. Communication cost is reduced. Rumour routing maintains only one path between source and destination as opposed to directed diffusion where data can be routed through multiple paths at low rates.

In the other hand, rumour routing does not perform well for a large number of events, the cost of maintaining agents and event-tables in each node becomes infeasible if there is not enough interest in these events from the BS.

4.2.2.1.4 Minimum Cost Forwarding Algorithm (MCFA)

Each node maintains the least cost path estimate from itself to the base-station, it does not need unique ID nor maintain a routing table, the direction of routing is always known.

The sensor node broadcasts the message to its neighbours. When these nodes receive the message, they check if it is on the least cost path between the source

sensor node and the base-station. If this is the case, it re-broadcasts the message to its neighbours. This process repeats until the base-station is reached.

4.2.2.1.5 Gradient-Based Routing

GBR [49] is a variant of Directed Diffusion.

The key idea is to memorize the number of hops when the interest is diffused through the whole network.

Each node can calculate a parameter called “the height of the node”, which is the minimum number of hops to reach the BS. The difference between a node's height and that of its neighbour is considered the gradient on that link. A packet is forwarded on a link with the largest gradient.

To uniformly divide the traffic over the network, some auxiliary techniques are used: data aggregation and traffic spreading.

In GBR, there are three different data dissemination techniques:

- Stochastic Scheme: A node picks one gradient at random when there are two or more next hops that have the same gradient.
- Energy-based scheme: A node increases its height when its energy drops below a certain threshold, so that other sensors are discouraged from sending data to that node.
- Stream-based scheme: New streams are not routed through nodes that are currently part of the path of other streams.

4.2.2.1.6 Information-driven sensor querying (IDSQ) and Constrained anisotropic diffusion routing (CADR)

On IDSQ [49] the querying node can determine which node can provide the most useful information with the additional advantage of balancing the energy cost. However, IDSQ does not specifically define how the query and the information are routed between sensors and the BS.

On the other hand, CADR [49] is a general form of Directed Diffusion. The key idea is to query only sensors that are close to a particular event and dynamically adjusting data routes.

The objective is to route data in the network such that the information gain is maximized while latency and bandwidth are minimized. The main difference from directed diffusion is the consideration of information gain in addition to the communication cost.

4.2.2.1.7 COUGAR

COUGAR [49] protocol views the network as a huge distributed database system. It uses in-network data aggregation to obtain more energy savings.

The sensor nodes select a leader node to perform aggregation and transmit the data to the BS. The BS is responsible for generating a query plan, which specifies the necessary information about the data flow and in-network computation for the incoming

query and send it to the relevant nodes. The query plan also describes how to select a leader for the query.

The main advantage of COUGAR is that by using data aggregation less energy is required.

However, the addition of query layer on each sensor node may add an extra overhead in terms of energy consumption and memory storage.

Also, to obtain successful in-network data computation, synchronization among nodes is required (not all data are received at the same time from incoming sources) before sending the data to the leader node.

Lastly, the leader nodes should be dynamically maintained to prevent them from being hot-spots.

4.2.2.1.8 Active query forwarding in sensor networks (ACQUIRE)

ACQUIRE [49] views the network as a distributed database where complex queries can be further divided into several sub queries.

The operation of ACQUIRE is as follows:

- The BS node sends a query, which is then forwarded by each node receiving the query. During this, each node tries to respond to the query partially by using its pre-cached information and then forwards it to another sensor node.
- If the pre-cached information is not up-to-date, the nodes gather information from their neighbours within a look-ahead of d hops. Once the query is resolved completely, it is sent back through the reverse or shortest-path to the BS.

The advantage of ACQUIRE is that it can deal with complex queries by allowing many nodes to send responses.

However, since the nodes become aware of events through the event agents, the heuristic for defining the route of an event agent highly affects the performance of next hop selection. The problem of selecting the next node for forwarding the query, which ACQUIRE addresses, has been studied in CADR and Rumor Routing .

4.2.2.1.9 Energy Aware Routing

The objective of Energy aware routing [49] is to increase the network lifetime.

It differs from DD in the sense that it maintains a set of paths instead of maintaining one optimal path at higher rates. These paths are maintained and chosen by means of a certain probability, whose value depends on how low the energy consumption of each path is.

The protocol initiates a connection through localized flooding, which is used to discover all routes between source/destination pair and their costs; thus building up the routing tables. The high-cost paths are discarded and a forwarding table is built by choosing neighbouring nodes in a manner that is proportional to their cost.

The main advantage of Energy Aware Routing is that longer network lifetime is achieved as energy is dissipated more equally among all nodes. The energy aware routing algorithm uses less energy than traditional algorithms for most realistic cases.

In the other hand, the guarantee of delivery of packets is improvable under situations where non-uniform transmission ranges exist. Furthermore, delays need to be decreased.

4.2.2.1.10 Routing Protocols with Random Walks

The objective of Routing protocols with random walks [49] is to achieve load balancing in a statistical sense and by making use of multi-path routing in WSNs. Only large scale networks where nodes have very limited mobility are considered.

It is assumed that sensor nodes can be turned on or off randomly, each node has a unique identifier but no location information is needed.

To find a route from a source to its destination, the location information is obtained by computing distances between nodes using the distributed asynchronous version of the well-known Bellman-Ford algorithm [58].

The main advantage of this technique is that nodes are required to maintain little state information. The problem is that the current algorithm is only valid for grid-topology sensor network.

4.2.2.2 Hierarchical Routing

In a hierarchical architecture, higher energy nodes can be used to process and send the information while low energy nodes can be used to perform the sensing in the proximity of the target.

Creating clusters and assigning special tasks to cluster heads can greatly contribute to overall system scalability, lifetime, and energy efficiency.

Hierarchical routing is mainly two-layer routing where one layer is used to select cluster-heads and the other layer is used for routing.

4.2.2.2.1 LEACH protocol

LEACH [49] is a cluster-based protocol which includes distributed cluster formation. LEACH randomly selects a few sensor nodes as cluster heads (CHs) and rotates this role to evenly distribute the energy load among the sensors in the network. The cluster head (CH) nodes compress data arriving from nodes that belong to the respective cluster and send an aggregated packet to the base station in order to reduce the amount of information that must be transmitted to the base station.

Data collection is centralized and is performed periodically. This protocol is most appropriate when there is a need for constant monitoring by the sensor network.

It has two phases:

- The setup phase: The clusters are organized and CHs are selected.
- The steady state phase: The data transfer to the BS takes place.

4.2.2.2.2 Power-Efficient Gathering in Sensor Information Systems (PEGASIS)

PEGASIS [49] is an enhancement over the LEACH protocol. In order to extend network lifetime, nodes need only communicate with their closest neighbours and they take turns in communicating with the base-station. When the round of all nodes communicating with the base-station ends, a new round will start and so on. This reduces the power required to transmit data per round as the power draining is spread uniformly over all nodes.

PEGASIS has two main objectives:

- Increase the lifetime of each node by using collaborative techniques and as a result the network lifetime will be increased.
- Allow only local coordination between nodes that are close together so that the bandwidth consumed in communication is reduced. Unlike LEACH, PEGASIS avoids cluster formation and uses only one node in a chain to transmit to the BS instead of using multiple nodes.

To locate the closest neighbour node in PEGASIS, each node uses the signal strength to measure the distance to all neighbouring nodes and then the node adjusts the signal strength so that only one node can be heard.

The chain in PEGASIS will consist of those nodes that are closest to each other and form a path to the base-station.

The main advantage of PEGASIS increases lifetime of the network twice that obtained for LEACH.

The problem is that it requires dynamic topology adjustment since a sensor node needs to know about energy status of its neighbours in order to know where to route its data. Such topology adjustment can introduce significant overhead especially for highly used networks. PEGASIS introduces excessive delay for distant node on the chain.

4.2.2.2.3 Threshold-sensitive Energy Efficient Protocols (TEEN and APTEEN)

TEEN and APTEEN [49] hierarchical routing protocols are proposed for time-critical applications.

TEEN

The sensor nodes sense the medium continuously, but the data transmission is done less frequently.

A cluster head sensor sends its members:

- Hard threshold (HT): Threshold value of the sensed attribute. The hard threshold tries to reduce the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest.

- Soft threshold (ST): Small change in the value of the sensed attribute that triggers the node to switch on its transmitter and transmit. The soft threshold further reduces the number of transmissions that might have otherwise occurred when there is little or no change in the sensed attribute. A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption.

Objective: Control the trade-off between energy efficiency and data accuracy.

The advantage of TEEN is its suitability for time-critical sensing applications. Since message transmission consumes more energy than data sensing, energy consumption in this scheme is less than the proactive networks.

However, if the thresholds are not received, the nodes will never communicate, and the user will not get any data from the network at all.

APTEEN

It is a hybrid protocol that changes the periodicity or threshold values used in the TEEN protocol according to the user needs and the type of the application.

In APTEEN, the cluster heads broadcast the following parameters:

1. Attributes (A): Set of physical parameters the user is interested about.

2. Thresholds: Hard Threshold (HT) and the Soft Threshold (ST).

3. Schedule: TDMA, assigning a slot to each node.

4. Count Time (CT): Maximum time period between two successive reports sent by a node.

The main advantage of APTEEN is that it combines both proactive and reactive policies. It offers a lot of flexibility by allowing the user to set the count-time interval (CT), and the threshold values for the energy consumption, which can be controlled by changing the count time as well as the threshold values.

The problem is that additional complexity is required to implement the threshold functions and the count time.

4.2.2.2.4 Small Minimum Energy Communication Network (MECN)

MECN protocol [49] computes an energy-efficient sub-network by using low power GPS.

It identifies a relay region for every node. The relay region consists of nodes in a surrounding area where transmitting through those nodes is more energy efficient than direct transmission.

The main idea is to find a sub-network, which will have less number of nodes and require less power for transmission between any two particular nodes. In this way, global minimum power paths are found without considering all the nodes in the network.

The main advantage of MECN is self-reconfiguring and thus can dynamically adapt to nodes failure or the deployment of new sensors.

SMECN [49] is an extension of MECN. In MECN, it is assumed that every node can transmit to every other node, which is not possible every time. In SMECN possible obstacles between any pair of nodes are considered.

4.2.2.2.5 Self Organizing Protocol (SOP)

A self-organizing protocol [49] and an application taxonomy are used to build architecture able to support heterogeneous sensors.

These sensors can be mobile or stationary. Some sensors probe the environment and forward the data to a designated set of nodes that act as routers. Router nodes are stationary and form the backbone for communication. Collected data are forwarded through the routers to the more powerful BS nodes. Each sensing node should be able to reach a router in order to be part of the network.

Local Markov Loops (LML) algorithm, which performs a random walk on spanning trees of a graph, was used to support fault tolerance and as a means of broadcasting. This algorithm incurs a small cost for maintaining routing tables and keeping a balanced routing hierarchy.

The main advantage of SOP is that the energy consumed for broadcasting a message is less than that consumed in the SPIN protocol.

4.2.2.2.6 Sensor Aggregates Routing

The objective of Sensor Aggregates Routing [49] is to collectively monitor target activity in a certain environment (target tracking applications).

A sensor aggregate comprises those nodes in a network that satisfy a grouping predicate for a collaborative processing task. Sensors in a sensor field are divided into clusters according to their sensed signal strength, so that there is only one peak per cluster. Then, local cluster leaders are elected. One peak may represent one target, multiple targets, or no target in case the peak is generated by noise sources. To elect a leader, information exchanges between neighbouring sensors are necessary. If a sensor, after exchanging packets with all its one-hop neighbours, finds that it is higher than all its one-hop neighbours on the signal field landscape, it declares itself a leader. This leader-based tracking algorithm assumes the unique leader knows the geographical region of the collaboration. Three algorithms were proposed:

- DAM protocol [49], which forms sensor aggregates for a target monitoring task.
- EBAM algorithm [49] estimates the energy level at each node by computing the signal impact area, combining a weighted form of the detected target energy at each impacted sensor assuming that each target sensor has equal or constant energy level.
- EMLAM algorithm [49] removes the constant and equal target energy level assumption.

4.2.2.2.7 Virtual Grid Architecture routing (VGA)

VGA [49] uses data aggregation and in-network processing to maximize the network lifetime.

Due to the node stationarity and extremely low mobility in many applications in WSNs, a reasonable approach is to arrange nodes in a fixed topology.

A GPS-free approach is used to build clusters that are fixed, equal, adjacent, and non-overlapping with symmetric shapes.

Inside each zone, a node is optimally selected to act as cluster head. Data aggregation is performed at two levels: local and then global. The set of cluster heads, also called Local Aggregators (LAs), perform the local aggregation, while a subset of these LAs is used to perform global aggregation.

The determination of an optimal selection of global aggregation points, called Master Aggregators (MAs), is NP-hard problem. The location of the base station is not necessarily at the extreme corner of the grid, rather it can be located at any arbitrary place.

4.2.2.2.8 Hierarchical Power-aware Routing (HPAR)

HPAR [49] protocol divides the network into groups of sensors. Each group of sensors in geographic proximity is clustered together as a zone and each zone is treated as an entity.

To perform routing, each zone is allowed to decide how it will route a message hierarchically across the other zones such that the battery lives of the nodes in the system are maximized. Messages are routed along the path which has the maximum over all the minimum of the remaining power, called the max-min path. The motivation is that using nodes with high residual power may be expensive as compared to the path with the minimal power consumption.

The max-min algorithm was proposed. This is based on the trade-off between minimizing the total power consumption and maximizing the minimal residual power of the network.

The algorithm tries to enhance a max-min path by limiting its power consumption as follows:

First, the algorithm finds the path with the least power consumption (Pmin) by using the Dijkstra algorithm. Second, the algorithm finds a path that maximizes the minimal residual power in the network.

4.2.2.2.9 Two-Tier Data Dissemination (TTDD)

TTDD [49] provides data delivery to multiple mobile base-stations.

Each data source proactively builds a grid structure which is used to disseminate data to the mobile sinks by assuming that sensor nodes are stationary and location-aware.

In TTDD:

- Sensor nodes: Stationary and location-aware.
- Sinks: May change their locations dynamically.

Once an event occurs, sensors surrounding it process the signal and one of them becomes the source to generate data reports.

To build the grid structure, a data source chooses itself as the start crossing point of the grid, and sends a data announcement message to each of its four adjacent crossing points using simple greedy geographical forwarding. During this process, each intermediate node stores the source information and further forwards the message to its adjacent crossing points except the one from which the message comes from. This

process continues until the message stops at the border of the network. After this process, the grid structure is obtained.

Using the grid, a BS can flood a query, which will be forwarded to the nearest dissemination point in the local cell to receive data. Then, the query is forwarded along other dissemination points up-stream to the source. The requested data then flows down in the reverse path.

The main advantage of TTDD is that it can achieve longer lifetimes than DD.

However, it has its disadvantages. For example, the length of a forwarding path in TTDD is always bigger than the length of the shortest path.

Longer data delivery delays than DD.

Overhead associated with maintenance and recalculation of the grid is high due to network topology changes.

TTDD assumes the availability of very accurate positioning system which is not yet available for WSNs.

4.2.2.3 Location based routing protocols

In this type of routing, sensor nodes are addressed by means of their locations. Relative coordinates of neighbouring nodes can be obtained by exchanging such information between neighbours.

Alternatively, the location of nodes may be available directly by communicating with a satellite, using GPS (Global Positioning System), if nodes are equipped with a small low power GPS receiver.

To save energy, some location-based schemes demand that nodes should go to sleep if there is no activity. More energy savings can be obtained by having as many sleeping nodes in the network as possible.

4.2.2.3.1 Geographic Adaptive Fidelity (GAF)

The network area is first divided into fixed zones and forms a virtual grid. Inside each zone, nodes collaborate with each other to play different roles. For example, nodes will elect one sensor node to stay awake for a certain period of time and then they go to sleep. This node is responsible for monitoring and reporting data to the BS on behalf of the nodes in the zone.

There are three states in GAF:

1) Discovery: Determining the neighbours in the grid.
2) Active reflecting participation in routing.
3) Sleep: When the radio is turned off.

The issue is how to schedule roles for the nodes to act as cluster heads. A cluster head can ask the sensor nodes in its cluster to switch on and start gathering data if it senses an object. Then, cluster head is responsible for receiving raw data from other nodes in its cluster and forward it to the BS.

The advantage of GAF is that it increases the lifetime of the network by saving energy.

4.2.2.3.2 Geographic and Energy Aware Routing (GEAR)

GEAR [49] uses energy aware and geographically-informed neighbour selection heuristics to route a packet towards the destination region. The key idea is to restrict the number of interests in directed diffusion by only considering a certain region rather than sending the interests to the whole network. By doing this, GEAR can conserve more energy than directed diffusion.

Each node in GEAR keeps an estimated cost and a learning cost of reaching the destination through its neighbours.

-Estimated cost: Combination of residual energy and distance to destination.

-Learned cost: Refinement of the estimated cost that accounts for routing around holes in the network. A hole occurs when a node does not have any closer neighbour to the target region than itself. If there are no holes, the estimated cost is equal to the learned cost.

There are two phases in the algorithm:

1) Forwarding packets towards the target region

2) Forwarding the packets within the region

4.2.2.3.3 MFR, DIR and GEDIR

MFR, DIR and GEDIR [49] protocols deal with basic distance, progress, and direction based methods. The key issues are forward direction and backward direction. A source node or any intermediate node will select one of its neighbours according to a certain criterion.

MFR

In most cases, MFR and Greedy methods have the same path to destination.

DIR

The best neighbour has the closest direction (that is, the lower angle) toward the destination. That is, the neighbour with the minimum angular distance from the imaginary line joining the current node and the destination is selected.

GEDIR

It is a greedy algorithm that always moves the packet to the neighbour of the current vertex whose distance to the destination is minimized. The algorithm fails when the packet crosses the same edge twice in succession.

4.2.2.3.4 The Greedy Other Adaptive Face Routing (GOAFR)

GOAFR [49] protocol combines greedy and face routing. The greedy algorithm of GOAFR always picks the neighbour closest to a node to be next node for routing.

FR

It is the first one that guarantees success if the source and the destination are connected. However, the worst-case cost of FR is proportional to the size of the network in terms of number of nodes.

AFR

It is the first algorithm that can compete with the best route in the worst-case. It is not average-case efficient.

OFR

It behaves well in dense networks, but it fails for very simple configurations.

GOAFR

It can achieve both worst-case optimality and average-case efficiency.

4.2.2.3.5 SPAN

SPAN [49] selects some nodes as coordinators based on their positions. The coordinators form a network backbone that is used to forward messages. A node should become a coordinator if two neighbours of a non-coordinator node cannot reach each other directly or via one or two coordinators (3 hop reachability).

4.2.3 Routing protocols based on protocol operation

4.2.3.1 Multipath routing protocols

Multipath routing protocols are a kind of routing protocols that use multiple paths rather than a single path in order to enhance the network performance.

The main advantage is that maintaining multiple paths allows a good fault tolerance. However this technique has many disadvantages, such as higher energy consumption and traffic generation. Network reliability can be increased at the expense of increased overhead of maintaining the alternate paths.

The idea is to split the original data packet into subpackets and then send each subpacket through one of the available multipaths. The studies show that, even if some of these subpackets are lost, the original message can still be reconstructed.

4.2.3.2 Query based routing

Query based routing is a technique in which the destination nodes propagate a query for data (sensing task) from a node through the network and a node having this data sends the data which matches the query back to the node, which initiates the query.

4.2.3.3 Negotiation based routing protocols

These protocols use high-level data descriptors in order to eliminate redundant data transmissions through negotiation.

The main idea of negotiation based routing in WSNs is to suppress duplicate information and prevent redundant data from being sent to the next sensor or the base-station by conducting a series of negotiation messages before the real data transmission begins.

4.2.3.4 QoS-based routing

In QoS-based routing protocols, the network has to balance between energy consumption and data quality.

In particular, the network has to satisfy certain QoS metrics, e.g., delay, energy, bandwidth, etc. when delivering data to the BS.

4.2.3.5 Coherent and non-coherent processing

Data processing is a major component in the operation of wireless sensor networks.

Two examples of data processing techniques:

a. Non-coherent data processing routing: Nodes will locally process the raw data before being sent to other nodes for further processing. The nodes that perform further processing are called the aggregators.
    1) Three phases in non-coherent processing:
    2) Target detection, data collection and pre-processing.
    3) Membership declaration.
    4) Central node election.
b. Coherent routing: The data is forwarded to aggregators after minimum processing. The minimum processing typically includes tasks like time stamping, duplicate suppression, etc. This is an energy efficient routing.

4.2.4 Comparison of WSN routing protocols

Table 4.3 show a comparison between the different WSN routing protocols commented.

Table 4.3: Classification and comparison of routing protocols in wireless sensor networks

	Classification	Power Usage	Negotiation-based	Data Aggregation	Localization	QoS	State Complexity	Scalability	Query-based
SPIN	Flat	Ltd	✓	✓			Low	Limited	✓
DD	Flat	Ltd	✓	✓	✓		Low	Limited	✓
RR	Flat			✓			Low	Good	✓
GBR	Flat			✓			Low	Limited	✓
MCFA	Flat						Low	Good	
CADR	Flat	Ltd		✓			Low	Limited	
COUGAR	Flat	Ltd		✓			Low	Limited	✓
ACQUIRE	Flat			✓			Low	Limited	✓
EAR	Flat						Low	Limited	✓
LEACH	Hirerarchical	Max		✓	✓		CHs	Good	
TEEN & APTEEN	Hirerarchical	Max		✓	✓		CHs	Good	
PEGASIS	Hirerarchical	Max			✓		Low	Good	
MECN & SMECN	Hirerarchical	Max					Low	Low	
SOP	Hirerarchical						Low	Low	
HPAR	Hirerarchical						Low	Good	
VGA	Hirerarchical		✓	✓	✓		CHs	Good	
SA	Hirerarchical			✓			Low	Good	Possible
TTDD	Hirerarchical	Ltd					Moderate	Low	Possible
GAF	Location	Ltd					Low	Good	
GEAR	Location	Ltd					Low	Limited	
SPAN	Location		✓				Low	Limited	
MFR, GEDIR	Location						Low	Limited	
GOAFR	Location						Low	Good	
SAR	QoS		✓	✓		✓	Moderate	Limited	✓
SPEED	QoS					✓	Moderate	Limited	✓

5. VANET simulators

5.1 Network Simulators

Since vehicular networks involve only wireless communications, all of the networking simulators described in this chapter support simulations with mobile wireless nodes.

To simulate a vehicular network, two different applications are needed, the mobility and the network simulator. A mobility simulator is generally used to produce node movement traces that are then fed to the network simulator. On the other hand, the network simulator then controls communications between the mobile nodes. As these network simulators support wireless communication, most of them include at least a simple node mobility model.

5.1.1 NS-2 and NS-3

In 1989, NS-2 [50] appeared as a network simulator that provided significant simulation of transport, routing, and multicast over wired and wireless networks. NS-2 was developed by the Information Sciences Institute of the University of Southern California.

Although the core of NS-2 is written in C++, but users interact with NS-2 by writing Tool Command Language (TCL) scripts. These scripts should contain all of the commands needed for specifying the simulation (e.g., setting up the topology, specifying wireless parameters, and so on).

There are implementations of several mobility models available for NS-2, including Random Trip Mobility [52] and Semi-Markov Smooth Mobility [53].

NS-2 is packaged with a bundle of rich libraries for simulating wireless networks. All the mobile nodes in NS-2 quickly assume that they are part of ad-hoc network and the simulation of mobile nodes connected with infrastructure networks is not really possible. To simulate a wireless node, the physical layer, the link layer and MAC (media access control) protocol are all included at the same time. But despite this, NS-2 is unable to simulate multiple radio interfaces. Moreover NS-2 has unrealistic models for wireless channel, which results in a biased radio propagation. For wireless

simulation, NS-2 supports only free space and two ray ground reflection models and cannot simulate path loss, multi-path fading and shadowing phenomena.

Besides, NS-2 only supports bi-directional (radiates or receives most of its energy in two directions) and omni-directional (radiates signal equally in all directions) antennas for signal propagation and waypoint mobility model for node movement.

When simulating wireless networks using NS-2, the nodes need to be programmed manually to sense and transmit data among each other. There is no built-in scanning facility available to sense other nodes that are floating around. Another constraint associated with NS-2 is that it cannot be extended to simulate a large mobile network.

However, some features made NS-2 to be the most widely used network simulator: it is open-source, many of the standard networking components and protocols are available and there is a well documented code base. On the other hand, this simulator is not easy to use because it has been extended by many developers so the architecture of NS-2 is very complex.

NS-3 [50] came around as a better replacement for its predecessor. It is written purely in C++ and limits the coding to only a few hundred lines as opposed to 300,000 lines for that of NS-2. For simulating huge networks NS-3 was equipped with support for distributed and federated simulation tasks.

5.1.2 GloMoSim and QualNet

GloMoSim (Global Mobile Information System Simulator) [50] is the second most popular network simulator after NS-2. It was developed by the Parallel Computing Laboratory in University of California, Los Angeles (UCLA) in 1999 and it was mainly targeted towards wireless network simulation and supporting the following mobility models: Random Waypoint [54], Random Drunken [55], and Trace-based [56]. Earlier GloMoSim had no support for GUI but now it includes a java based front end as well.

This simulator was coded in Parsec and all new protocols must be defined in Parsec as well. Parsec is a C-based simulation language, developed by the Parallel Computing Laboratory at UCLA, for sequential and parallel execution of discrete-event simulation models. It can also be used as a parallel programming language.

Another feature is that it has the ability to run on SMP (Shared-Memory symmetric Processor), which allows that memory is simultaneously accessible by all programs and helps to divide the network into separate modules each running as a distinct process. This reduces the load on CPU by dividing its workload. Because of these multi-tasking features, the user is able to simulate tens of thousands of nodes in a single simulation.

This simulator is packaged with libraries for simulating multiple mobility models including Random Drunken (template for designing new mobility models and node chooses its direction from four choices to choose a path randomly) and Trace-based models (model provided by user) apart from the Random Waypoint mobility model. Regarding radio propagation models, GloMoSim supports two-ray and free-space.

GloMoSim was designed to support millions of nodes in a single simulation using parallelism techniques. Node aggregation has always been the bottleneck in most simulations but this simulator was the first one to implement it succesfully. It also has the ability to support layer aggregation.

The input files of this simulator are nodes.input, mobility.in, app.conf, and config.in. The main configuration parameters for setting up a scenario are defined in config.in, which specifies the mobility, simulation time, radio-related parameters, all layers' protocols, and the application configuration. The file node.input specifies the nodes' topology (coordinates of each node). The file app.conf specifies the applications that generate traffic, and mobility.in specifies the trace, or trip, of nodes. An output file GLOMO.STAT is generated at the end of a simulation, which contains statistical information for each node at a certain layer.

GloMoSim version 2.0 was released in 2000 and after that, PARSEC stopped working on freeware software and released a commercial version of GloMoSim called QualNet [50].

QualNet includes a sophisticated GUI for setting up and running simulations, and provides a large set of wireless physical and MAC layer models. It also includes the following mobility models: group mobility, pedestrian mobility, and Random Waypoint. QualNet is available for both Windows and Unix/Linux platforms.

5.1.3 OPNET

OPNET [50] is a commercial network simulator mainly used to simulate wired and wireless communications networks. It supports a wide range of wireless technologies such as MANETs, IEEE 802.11 WLANs, WiMAX, Bluetooth, and satellite networks.

This simulator provides a graphical editor interface to build models for various network entities from physical layer modulator to application processes, and it includes graphical packages and libraries for presenting simulation scenarios and results.

There are three basic phases of the OPNET deployment process. First, choose and configure node models to use in simulations, such as a wireless node, trajectory, and so on. Second, build and organize the network by setting up connections for different entities. Third, select the desired statistics (local or global) to collect during the simulation.

OPNET includes only simple mobility models such as Random Drunken Model, Random Waypoint Model and Trace (vehicle movement is based on an imported trace file).

The scenario file describes network parameters, mobility parameters, and global parameters (e.g., simulation time). A configuration file converter retrieves specific elements from the network and node configuration files and reorders these elements into a standard (default) configuration format. The network configuration file specifies all of the network related parameters, for example, nodes, routers, applications, protocols, radios, and so on. The node configuration file includes the nodes' identities (such as IP addresses). The global parameters define the simulation time, coordinate system, random number seed, file-based node placement, protocol stack, statistic filter, and so on. In addition to the GUI-based output, the simulation produces a statistics file, which contains all the statistics generated from the simulation run.

5.1.4 OMNeT++

OMNeT++ [50] is an open-source simulation environment. The primary simulation applications are Internet simulations, mobility, and ad-hoc simulations. This simulator has a component-based design, meaning that new features and protocols can be

supported through modules. OMNeT++ supports network and mobility models through the independently developed Mobility Framework and INET Framework modules.

OMNeT++ simulations consist of simple modules that implement the atomic behaviour of a model, e.g. a particular protocol. Multiple simple modules can be linked together and form a compound module. This linking and the set-up of the network simulation takes place in NED, OMNeT++'s network description language. NED is transparently rendered into C++ code when the simulation is compiled as a whole. Moreover, NED supports the specification of variable parameters in the network description: for example, the number of nodes in a network can be marked to be dynamic and later on be configured at runtime.

Simulation design in OMNeT++ is GUI-based, and output data can be plotted through the GUI as well. OMNEST is the commercial version of OMNeT++.

5.1.5 J-Sim

J-Sim [50] is an open-source simulation environment, developed entirely in Java. This simulator provides two mobility models: trajectory-based and random waypoint. J-Sim is presented as an alternative to NS-2, because it is designed to be easier to use.

In J-Sim, applications are built as a set of components that can be designed and tested separately. J-Sim can take a TCL file as input, similar to NS-2, but with a different format. Like NS-2, this simulator produces an event trace file and an animation file.

5.1.6 SWANS

SWANS (Scalable Wireless Ad-hoc Network Simulator) [50] was developed to be a scalable alternative to NS-2 for simulating wireless networks. SWANS is organized as independent software components that can be composed to form complete wireless network or sensor network configurations.

Based on comparisons of SWANS, GloMoSim, and NS-2, SWANS was determined to be the most scalable and the most efficient in memory usage with the fastest runtime. Along with better performance, SWANS delivered similar results as NS-2, at least for the network components that were implemented in both.

The input for SWANS is a Java file that creates the nodes and specifies how these nodes should move (the node movement scenario) and how they should communicate (the communication scenario). The user can select any of the ready-made applications in SWANS and associate it with any node(s) to execute it at the node application layer. Also, SWANS gives the user the flexibility to build a custom application and execute it at the application layer of any node.

5.2 Tightly Integrated Simulators

As mentioned before, the simulation of VANET applications not only requires simulating the wireless communication between the vehicles, but also requires simulating the mobility of the vehicles. Unfortunately, these two aspects of VANET simulation have often been decoupled.

The problem is how to merge the two types of simulators (network simulator and mobility simulator). A simple method to achieve this merge and create a tightly integrated simulation is to implement mobility models in a network simulator, but

without allowing the network messages to feed back to the mobility model. This kind of simulation is called one-way communication (from mobility model to network). These types of simulators are suitable for simulating infotainment-related VANET applications, including Internet connectivity, multimedia applications, and peer-to-peer applications, where the communication does not affect the movement of the vehicles.

In contrast, tightly integrated simulators that offer two-way communication usually consist of two sub-simulators (network and mobility) that can communicate with each other. These simulators are more appropriate for safety-related and traffic information applications that assume that feedback from the network will affect vehicles movements.

In these types of applications, the traffic simulator feeds the network simulator with position information, speed, acceleration, direction, and so on. The VANET application that runs at the top level of the network simulator incorporates this information with surrounding vehicles information in order to notify the driver of upcoming congestion or a possible collision. Based on this notification, driving decisions (i.e., vehicle mobility) may be affected. For example, in a congestion notification system, the driver may choose to change lanes or take a different path. These decisions need to propagate back to the mobility simulator to be reflected in the vehicle mobility information.

Usually, any simulator has an events queue to store the events that should be executed according to their scheduled execution time. In the case of two-way communication simulators, each sub-simulator has its own events queue. These two events queues can be combined into one events queue, or they can be kept separate, which implies that extra overhead will be needed for synchronization. Based on this decision, the two-way communication simulators are separated into two categories: those with a single events queue and those with two events queues.

Having a single events queue can be achieved through implementing one of the sub-simulators in the other. Often, the vehicular mobility sub-simulator is implemented in the network sub-simulator, as in ASH (Application-aware SWANS with Highway mobility), or the two simulators can be highly integrated together, as in NCTUns.

The advantages of having a single events queue are that the vehicles mobility events and the network events will be inserted in the same queue, which removes the burden of synchronizing the two types of events. In addition, the simulation will be more efficient from the execution time and memory consumption perspectives. The main disadvantage of having a single events queue is that the process of maintaining and extending such simulators is not easy.

With two events queues, two-way communication is achieved through an interface that is implemented between the network sub-simulator and the mobility sub-simulator.

The main function of that interface is to update each sub-simulator with the recent events in the other sub-simulator. Also, it synchronizes the event execution in each of the events queues. The main disadvantages are that these types of simulators consume more memory and execution time.

5.2.1 SWANS++

SWANS++ [50] extends the network simulator SWANS by adding a GUI to visualize the scenario and a mobility model, STRAW (STreet RAndom Waypoint), for the vehicles movement in street scenarios. STRAW uses the simple random waypoint mobility model, but it restricts the vehicles movement to real street boundaries, loaded

from TIGER/Line data files [57]. TIGER/Line data files contain features such as roads, railroads, rivers, as well as legal and statistical geographic areas.

STRAW consists mainly of three components: intra-segment mobility, intersegment mobility, and route management and execution. In intersegment mobility, the vehicles move according to a car-following model and change their speed only in certain situations. For intersegment mobility, according to the system design, there is either a traffic control sign or a stop sign at each intersection that forces the vehicle to alter its speed. The mobility model implemented in STRAW (and therefore, SWANS++) does not support lane changing. The route management and execution (RME) module is responsible for determining the vehicles routes during the simulation.

SWANS++ is a tightly integrated simulator, but it does not provide feedback between the mobility and networking modules.

5.2.2 GrooveNet

GrooveNet (originally known as GrooveSim) [50] is an integrated network and mobility simulator that allows communication between real and simulated vehicles. Originally, GrooveNet extended the open-source simulator roadnav by adding a network model and a GUI based on Qt. GrooveNet can load real street maps from the TIGER/Line database [57] in order to simulate vehicles mobility on real roads. GrooveNet includes fixed mobility, street speed, uniform speed, and car-following mobility models. This simulator supports many operational modes, including drive mode, simulation mode, playback mode, hybrid simulation mode, and test generation mode.

GrooveNet's unique ability to integrate simulated vehicles with real vehicles allows it to function as testbed software as well as a simulator.

5.2.3 TraNS

TraNS (Traffic and Network Simulation Environment) [50], which was presented at MobiCom 2007,can be considered as the first pure VANET simulator. It was the first work to combine a network simulator, NS-2, with a vehicular mobility simulator: SUMO, and to provide feedback from the network simulator to the mobility simulator. TraNS can operate in two modes: network-centric mode and application-centric mode. In the network-centric mode, there is no feedback provided from NS-2 to SUMO, so the vehicles mobility trace file can be pregenerated and fed to the network simulator later. The link between the two simulators in this case is done through a parser that analyzes the mobility trace file generated by SUMO and converts it to a suitable format for NS-2. In the application-centric mode, the feedback between NS-2 and SUMO is provided through an interface called TraCI. In this mode, the two simulators (SUMO and NS-2) must run simultaneously. TraCI achieves the link between NS-2 and SUMO by converting the mobility commands coming from NS-2 to a sequence of mobility primitive commands such as stop, change lane, change speed, and so on that can be sent to SUMO. As both simulators are running separately at the same time, the two-way communication in application-centric mode uses two separate events queues.

5.2.4 Veins

Veins (Vehicles in Network Simulation) [50] is another simulator that couples a mobility simulator with a network simulator. In Veins, SUMO is paired with OMNeT++ by extending SUMO to allow it to communicate with OMNeT++ through a TCP connection.

In order to create a bidirectional communication between the two simulators, OMNeT++ has also been extended by adding a module that allows all participating nodes (vehicles) to send commands via the established TCP connection to SUMO. In this case, the two extensions represent the interface between the network simulator and the mobility simulator. Thus, the network simulator can react to the received

mobility trace from the mobility simulator by introducing new nodes, by deleting nodes that have reached their destination, and by moving nodes according to the instructions from the mobility simulator.

In Veins, there is a manager module that is responsible for synchronizing the two simulators. Thus, as with TraNS, this simulator has two separate queues of events.

5.2.5 NCTUns

NCTUns (National Chiao Tung University Network Simulator) [50] implements two-way communication with a single events queue. NCTUns 1.0 was developed only as a network simulator, but the most recent version, NCTUns 6.0, integrates some traffic simulation capabilities, such as designing maps and controlling vehicles mobility. Also, NCTUns includes a GUI to aid in the design process of the maps.

The supported vehicular movement has two modes, pre-specified and autopilot. In the pre-specified movement mode, the scenario designer specifies the moving path and the speed for each vehicle. In autopilot mode, the scenario designer specifies the following parameters for each vehicle: initial speed, maximum speed, initial acceleration, maximum acceleration, maximum deceleration, and so on. Then, the autopilot selects the best route to navigate in the map and it is also capable of performing car following, lane changing, overtaking, turning, and traffic light obeying.

The NCTUns network simulator has many useful features: it directly uses the real-life Linux TCP/IP protocol stack to generate high-fidelity simulation results, it provides a highly-integrated and professional GUI environment and its simulation engine adopts an open-system architecture and is open source. Moreover, it simulates various important networks: Ethernet-based fixed Internet, IEEE 802.11(b) wireless LANs, mobile ad-hoc (sensor) networks, GPRS cellular networks, optical networks (including both circuit-switching and burst-switching networks), IEEE 802.11(b) dual-radio wireless mesh networks, IEEE 802.11(e) QoS wireless LANs, Tactical and active mobile ad-hoc networks, IEEE 802.16(d) WiMAX wireless networks (including the PMP and mesh modes), DVB-RCS satellite networks, wireless vehicular networks for Intelligent Transportation Systems (including V2V and V2I), multi-interface mobile nodes for heterogeneous wireless networks, IEEE 802.16(e) mobile WiMAX networks, IEEE 802.11(p)/1609 WAVE wireless vehicular networks, various realistic wireless channel models, IEEE 802.16(j) transparent mode and non-transparent mode WiMAX networks, etc.

On the 1st of December of 2010, the commercial version of NCTUns was announced to all the users of the simulator via e-mail: it will be called EstiNet. According to its developers, EstiNet is launched to provide better quality, functionality, performance and support.

5.2.6 Gorgorin et al.

Gorgorin et al. [50] developed an integrated vehicular network simulator that allows feedback between the network and mobility modules using a single event queue. Map

information is imported from the TIGER/Line database [57], but because the information in these maps does not include the number of lanes or traffic control information, the authors augment the maps with this missing information heuristically.

For the vehicular mobility model, the authors implemented a microscopic traffic simulator that is based on the driver's behaviour, assuming that the driver will be in one of the following four modes: free driving, approaching, following, or braking. The driver's behaviour is determined based on the distance to the preceding vehicle. Not all drivers have the same personality, so the simulator allows for different driver profiles (i.e., aggressive, regular, and calm). The mobility model implemented has been validated against traces taken from a German freeway and a U.S. freeway.

For the network simulator, the authors implemented a physical layer that depends on both cumulative noise calculation and signal to noise ratio (SNR) to determine whether to accept or drop the received packet. Also, they have implemented CSMA/CA to represent the MAC layer. For the routing layer, they have implemented a geographical routing protocol. The new feature in this simulator is that the authors implemented a model to compute the fuel consumption and pollutant emissions in order to find the relationship between these measurements and the vehicle's speed and acceleration.

5.2.7 ASH

ASH (Application-aware SWANS with Highway mobility) [50] is an extension of the wireless network simulator SWANS that implements the IDM vehicular mobility model and MOBIL lane changing. ASH supports feedback between the vehicular mobility sub-simulator and the network sub-simulator, making it one of the two-way communication simulators with a single events queue. ASH allows users to design a simple highway segment and customize it by specifying the directions (one-way or two-way), the number of lanes, the number of entries and exits and their corresponding locations along the segment.

In addition to adding highway mobility models to SWANS, ASH extends the node types available:

1. Mobile Communicating Node. This represents a participating vehicle that should execute a user-defined application, which specifies how the vehicle should behave.
2. Mobile Silent Node. This represents a non-participating vehicle that should execute a null application so that it will not be able to send or receive any messages.
3. Static Communicating Node. This represents road-side infrastructure that should execute a user-defined application, which specifies how the road-side unit should behave. Also, this kind of node may have different physical layer characteristics (e.g., transmission power) than the mobile communicating nodes.
4. Static Silent Node. This represents a road obstacle that should execute a null application.

Because most VANET applications use flooding-based techniques to disseminate data, ASH also implements the Inter-Vehicle Geocast protocol (IVG). Moreover, it supports a probabilistic version of IVG to take the surrounding traffic density into account.

ASH accepts a configuration file for the highway scenario. The nodes creation and the communication scenario should be specified in a Java file as done in SWANS.

5.3 Conclusions about VANET simulators

Of all the above simulators, NCTUns was chosen to perform the tests of the project. The main reasons to use NCTUns were the fact that is a free of use open-source tool and its important capabilities for wireless vehicular network research. Moreover, the user does not need to deal with complex coding thanks to the powerful GUI support provided by this simulator. Using the GUI tool, vehicles can be deployed automatically.

NCTUns developers showed interest in Intelligent Transportation Systems (ITS) since version 4.0 when traffic and network simulators were coupled inside a single module to provide a single vehicular environment. With version 4, some important features for ITS simulation were added:

- Driver behaviour model
- Network road construction
- RSU and OBU simulation
- Car agent module: With its intelligent driving behaviour the car agent can model a car to obey certain parameters like traffic light, nearby vehicle, changing the lane, taking the turn and car following model.

In Table 5.1, motion level features of GloMoSim, NS-2, NCTUns and QualNet are presented. NCTUns offers a wide range of configurations regarding signal reception, Fading model, Path Loss model, wireless technology and type of antenna. On the other hand, it cannot manage a network with a large number of nodes.

Table 5.1: Motion level features of various network simulators

Mobility Model Class	GloMoSim	Ns-2	NCTUns		QualNet
Signal to noise ratio calculation	Cumulative	Difference in two signals	Cumulative		Cumulative
Signal reception	SNRT, BER	SNRT	Sender Transmitting power	Receiver Power threshold, Distance	SNRT, BER
Fading	Rayleigh, Ricean	No	Rayleigh, Ricean		Rayleigh, Ricean
Path loss	Free space, Two ray	Free space, Two ray	Free space, Two ray, Free space with shadowing		Free space, Two ray, ITM (Irregular Terrian Model)
Support for multiple wireless technology	Yes	No	Yes		Yes
Antennas's support	Bi-directional, Omni-directional	Bi-directional, Omni-directional	Directional, Bi-directional, Rotating		Bi-directional, Omni-directional, Switched
Random Movement	Yes	No	Yes		Yes
Time required for simulating 5000 Nodes (sec)	6191	Fail	Fail		6191
Memory Required for simulating 5000 Nodes (KB)	27.5	Fail	Fail		27.5
GUI	Yes	No	Yes		Yes

Table 5.2: Benefits and drawbacks of several VANETs simulator.

Mobility Model Class	Integrated Framework Support	Benefits	Drawbacks
Random Movement	Virtually All	- Straightforward, intuitive - Readily available	- Imprecise - Potentially unstable
Real-World Traces	GloMoSim, QualNet, OPNET, ns-2, Shawn, JiST/SWANS, OMNet++/INET Framework	- Most realistic node movement - Re-usable traces	- Costly and time-consuming - No free parameterization
Artificial Mobility Traces	GloMoSim, QualNet, OPNET, ns-2, Shawn, JiST/SWANS, OMNet++/INET Framework	- Realistic node movement - Free parameterization - Re-usable traces	- No feedback on driver behaviour
Bidirectionally Coupled Simulators	Ongoing efforts for: ns-2, Shawn, JiST/SWANS, OMNet++/INET Framework	- Realistic node movement - Free parameterization - Feedback on driver behaviour	- No re-usable traces

The main benefits and drawbacks of some VANET simulators are presented in Table 5.2.

According to Fig. 5.1, NCTUns is showed as a complete simulator which comprises both network and traffic simulator.

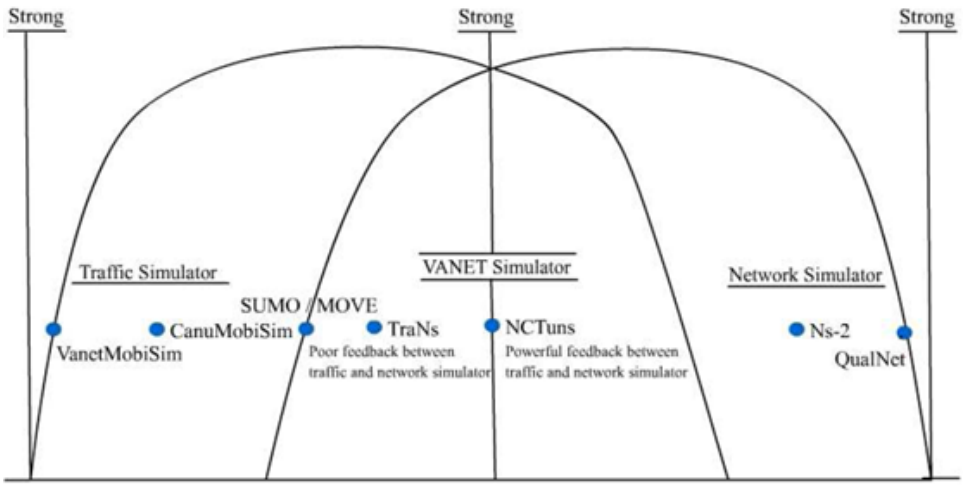


Fig. 5.1: Strength of traffic.

6. Simulations and results

6.1 The NCTUns simulator

Understanding how vehicular networks work and how they perform on a high-speed scenario was also a goal of this project. Hence, some simulations were performed using the NCTUns simulator that was introduced in the previous chapter. The simulations were run using the 6th version of this simulator, in particular the January 2010 release. In this section are described the installation process and the design of the simulation scenario, as well as the analysis of the results obtained.

6.1.1 Computing environment used

Instead of working with a native installation of the simulation system, the team worked with virtual machines. The main reason for that is the broad spectrum of machines used in the team, most of them laptops, which present some driver incompatibilities with the Linux distribution supported by NCTUns, i.e. Fedora Core.

The use of virtual machines also allows to easily deploying additional simulation environments, for example, for running multiple parallel simulations. Also, with virtual machines the assigned resources can be increased at need.

Another advantage is the possibility of creating a backup of the simulating environment without difficulties. The main disadvantage is a little decrease of computing power that is not very significant.

In the development of this project, VmWare desktop virtualization solutions were used. Specifically, "VmWare Player" for Windows and Linux environments and "VmWare Fusion" for MacOS X. There are also other free virtualization solutions such as Sun's Virtualbox, but it showed performance problems with the Fedora Core distribution.

6.1.2 Installation of the simulator

6.1.2.1 Previous considerations

Each release of NCTUns is developed for a specific version of Fedora Core. The reason of this requirement is the modifications that the simulator installer does on certain parts of the Linux kernel. Obviously, in order to avoid a simulator malfunction, the system kernel must not be updated from the one that comes by default with the Fedora Core release.

6.1.2.2 Installation steps

NCTUns is a free and open source software that can be downloaded at the following url: *http://nsl.csie.nctu.edu.tw/* .

Before installing it, some dependencies must be installed:

```
# yum install gcc-c++ readline-devel rsh-server xinetd tclreadline-devel
```

Then, we decompress the NCTUns distribution and execute the install.sh script. All the software is installed at the /usr/local/nctuns folder and the tunnel interfaces are created in the /dev directory.

During the process, the installer asks if a nctuns user should be created, if the kernel has to be patched and if the SELinux should be desactivated. In this installation all these questions were answered with yes.

Once the installation is finished, the system must be rebooted and started with the patched kernel by selecting the “NCTUNS” option in the GRUB bootloader.

The last step is adding the following environment variables to the `.bashrc` file:

```
    export NCTUNSHOME=/usr/local/nctuns
    export NCTUNS_TOOLS=$NCTUNSHOME/tools
    export NCTUNS_BIN=$NCTUNSHOME/bin
    export PATH=${PATH}:${NCTUNS_BIN}
```

After all these steps, we will be able to execute the simulator client, `nctunsclient` .

6.1.2.3 Starting NCTUns

As it was previously commented, NCTUns comprises two more components apart from the client, the coordinator and the dispatcher. These components have to be executed as root, and although it is not necessary to execute them at the same machine as the client, it is advisable.

The best way to execute these components is to execute it in different consoles, since each of them echoes information about the simulation that can be useful to debug if there is a problem.

Once the coordinator and the dispatcher are up, the `nctunsclient` can be started.

6.1.3 Design of an example VANET scenario in NCTUns 6.0.

NCTUns uses a simple and effective syntax and can be entirely configured through the graphical interface. Starting from a blank scenario in the NCTUns client, the steps to follow to design a VANET scenario like the one used in the simulations are described below.

1. To start the design of the scenario it is necessary to enter the “Draw topology” mode (Fig. 6.1). Then, to draw the road, the “ITS road segment” is selected (Fig. 6.2)

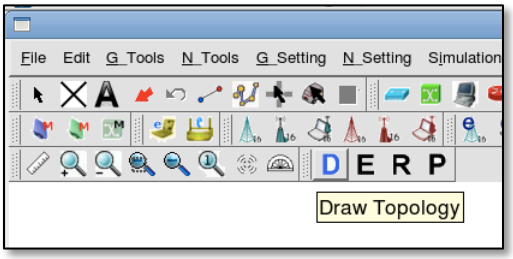


Fig. 6.1: Selecting the “Draw topology” mode.

Fig. 6.2: Drawing the road.

2. When drawing the road it is important to remember that the path must be closed and it must include an intersection because otherwise the cars would not move. Fig. 6.3 shows an example.

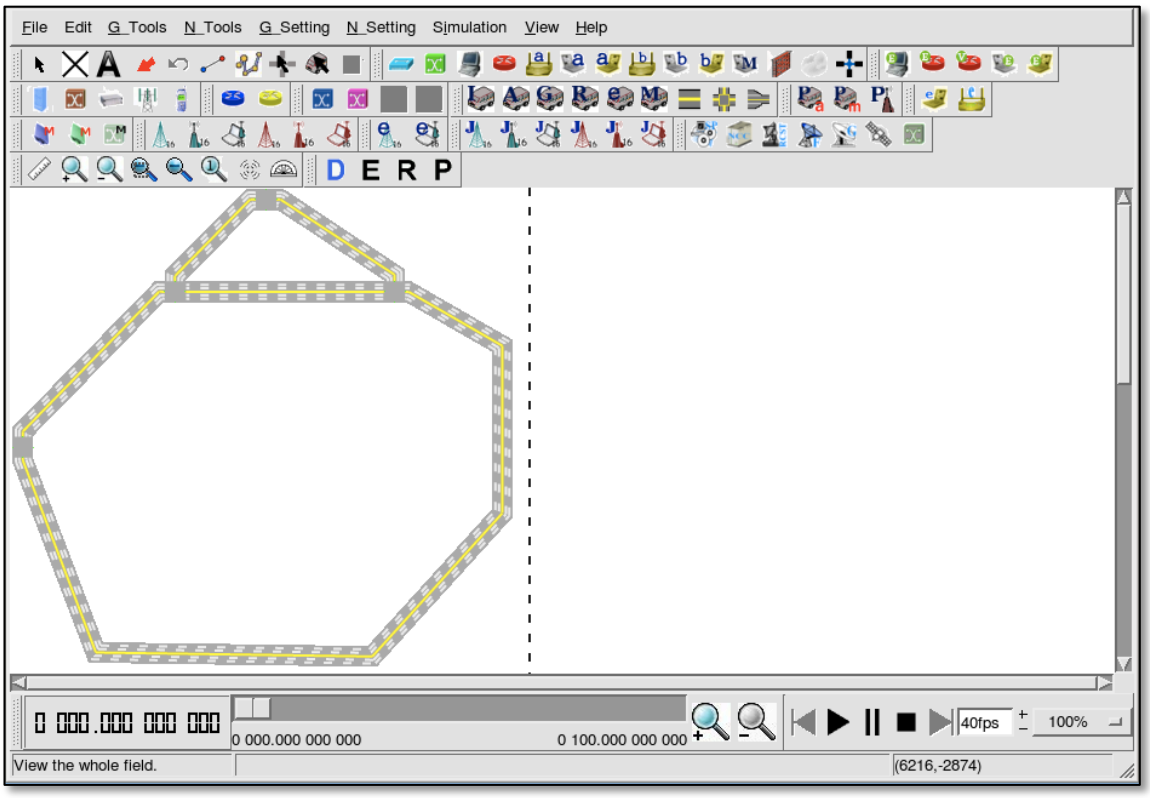


Fig. 6.3: Screenshot of the scenario.

3. Then, some RSUs and OBUs must be placed to populate the scenario. In this project the RSUs are of the kind “802.11(b) mobile node (ad-hoc mode)” (Fig. 6.4) and the OBUs are “802.11(b) ad-hoc mode interface” (Fig. 6.5).

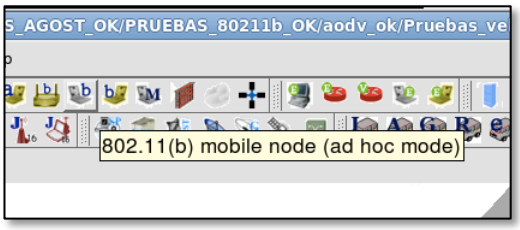


Fig. 6.4: Type of RSU selected.

Fig. 6.5: Type of OBU selected.

4. Once the different nodes are placed, switching to the “Edit property” mode (Fig. 6.6) allows setting the network parameters of each node.

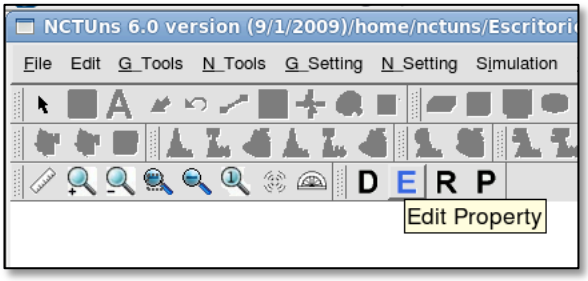


Fig. 6.6: Setting the “Edit Property” mode.

Double clicking on an OBU or RSU opens the “Mobile station” menu. The “Node editor” button in the “Path” tab allows modifying the protocol stack of the network interface (Fig. 6.7). Initially, the nodes feature but in this project it has been replaced for AODV.

In the “Application” tab we can configure the programs that will generate the traffic and the mobility model of the OBUs. NCTUns features different built-in applications, which use can be consulted by clicking in the “App. Usage” button.

It is also possible to develop an application and communicate it with NCTUns. This process is described in [53] and [54].

In the simulations of this project the network traffic is generated by the stg command, which generates a constant bit-rate flux. The receivers use the rtg application. On the other hand, the mobility model of the OBUs is generated by the “CarAgent”.

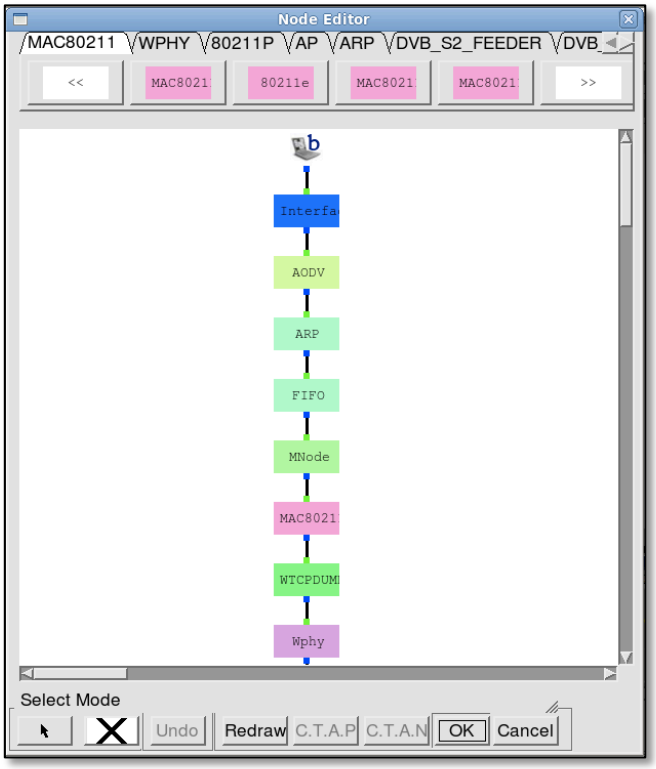


Fig. 6.7: The node editor.

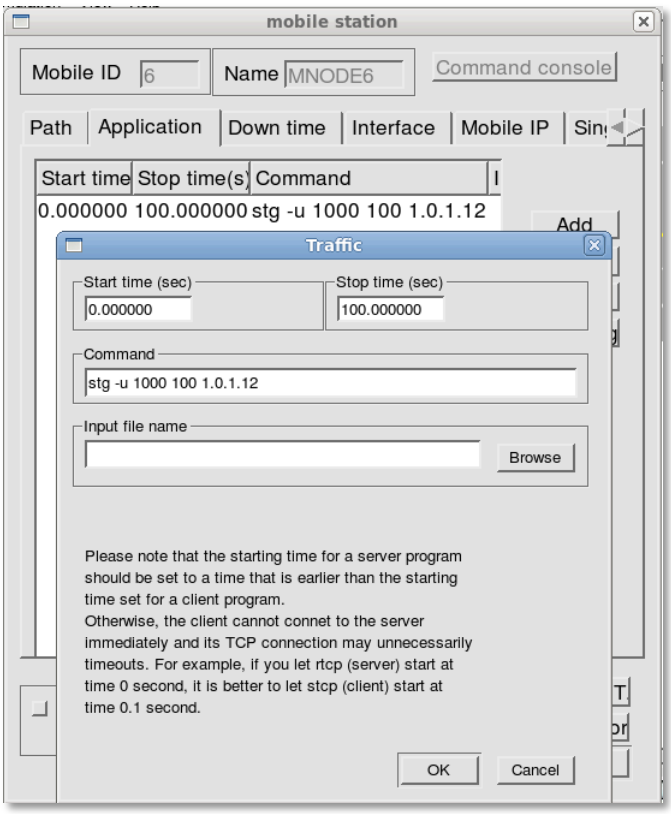


Fig. 6.8: The application configuration formulary.

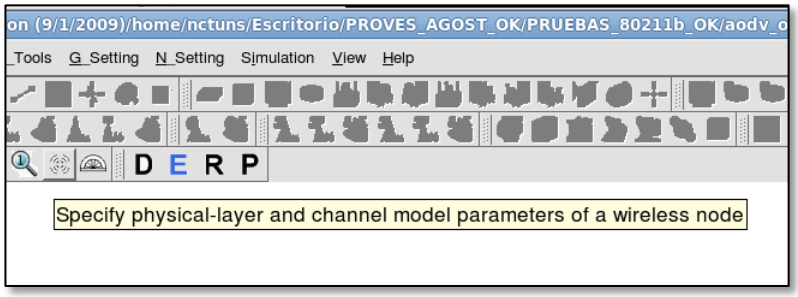


Fig. 6.9: Opening the Physical Layer editor.

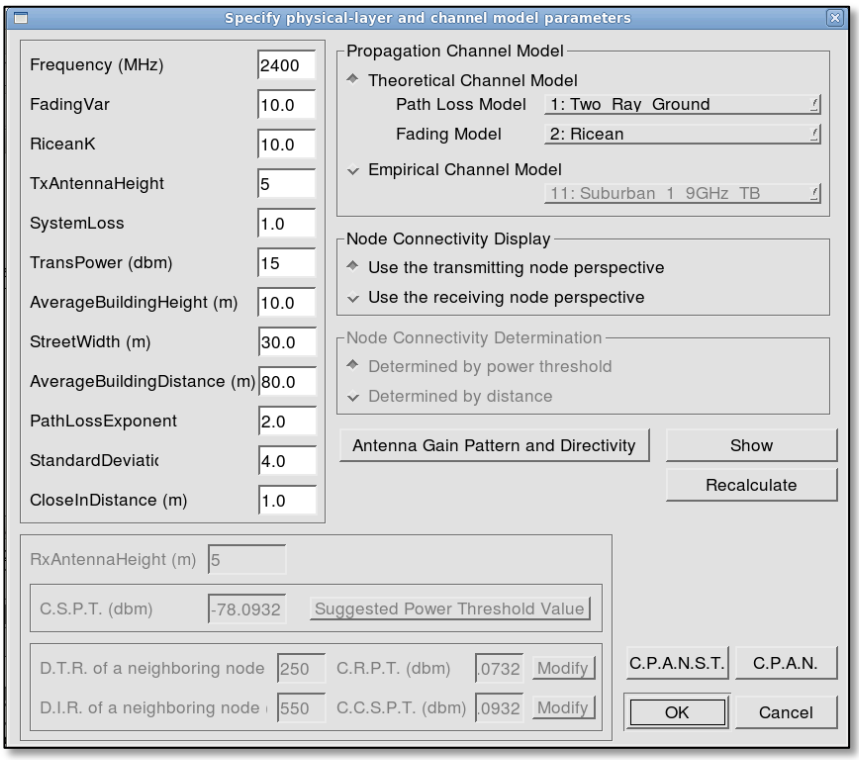


Fig. 6.10: The physical layer editor.

For RSUs, different parameters have to be set. The antenna's height is 5 meters, the “Path Loss Model” is “Two Ray Ground” and the “Fading Model” is “Ricean”. The Node Connectivity Display uses the receiving node perspective and the Node Connectivity is determined by distance. For all the tests, DTR is set to 120 meters and DIR to 240 meters for both RSUs.

On the other hand, OBUs have an antenna height of 1 meter. In the tests, there were two configurations:

- DTR: 75 meters and DIR: 150 meters
- DTR: 100 meters and DIR: 200 meters.

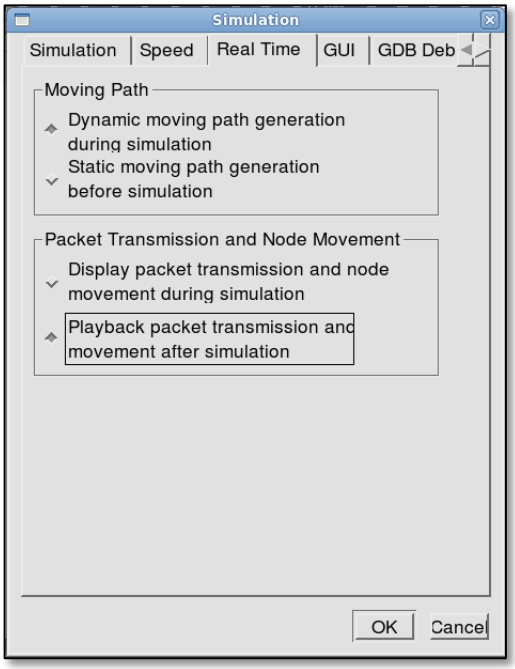


Fig. 6.11: Editing the simulation parameters.

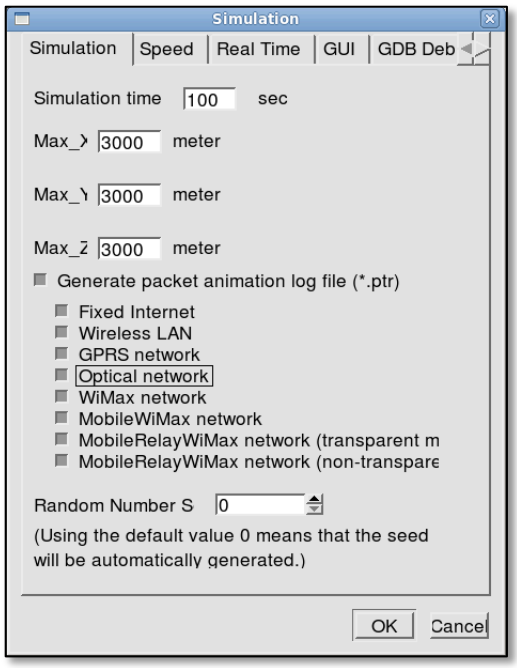


Fig. 6.12: Editing the simulation parameters. (II)

Opening “G_Settings” > “Simulation” menu (Fig. 6.11 and Fig. 6.12), allows editing the simulation parameters. On Real Time tab, the Dynamic moving path is set as well as the playback packet transmission after simulation. This allows observing the network communications on the replay.

The simulator measures time in “tics”. In the “Speed” tab we can configure the duration of one tic. Smaller durations will increase the simulation time.

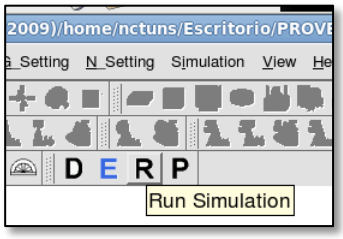


Fig. 6.13: Setting the “Run Simulation” mode.

Once in the “Run simulation” mode (Fig. 6.13), the topology is saved and several files are generated. The simulation is started with the “Simulation” > “Run” option.

6.1.4 Filtering the traces

NCTUns generates a binary trace file, which the GUI client reads to do the replays. The printPtr application allows decoding this binary file into plain text.

```
$ printPtr trace_file.ptr > plain_trace_file.tr
```

In this project, two types of studies have been performed. The first is a global study of three metrics, the percentage of lost packets, the throughput and the end-to-end delay.

The second study focuses in a individual study of certain cases to understand what is happening through time. Some of the metrics obtained in this second study are transmitted and received packets, collisions or length of the routes computed by the routing protocol.

Multiple filters were developed to obtain all these metrics from the traces. These filters have are coded in AWK [55] and shell script.

The code of one of the filters is shown below. This filter calculates the global metrics of each simulation (losses, delay and throughput).

```
BEGIN{
    # output: file or device where the output results will be redirected.
    output="/dev/stdout"
    # size: Size of packet in bytes.
    size=500
    # tics_per_second: Number of tics are in one second.
    # For our simulation, 1tic = 100ns
    tics_per_second=1e7
    # simulation_time: Length of the simulation.
    simulation_time=80
    # received: Count of the received packets that follow the restrictions.
    received=0
    # transmitted: Count of the transmitted packets that
    # follow the restrictions.
    transmitted=0
    # total: Summatory of the delay times.
    total=0
    # tx: Count of packets transmitted and received.
    tx=0
}

{
    # Mapping of the trace parameters to variables.
    # <protocol> <event type> <time> <duration> <packet type>
    # <source/destination> <tx rx> <packet ID> <packet length>
    # <count of re-tx> <drop reason> <freq. channel>
    # event: Event type.
    event_type=$2
    # packet_type: Packet type.
    packet_type=$5
    # source: Node ID that starts the transmission.
    source=$8
    # destination: Node ID to whom the packets are addressed.
    destination=$10
    # start_time: Time (in tics) when the transmission is started.
    start_time=$3
    # packet_id: Packet's unique id.
    packet_id=$11
    # packet_length: Length of the packet (counting the headers).
    packet_length=$12

    if (packet_type == "DATA" && event_type == "RX" && destination == "26>"
&& packet_length == "570")
    {
        received++
        # startRX: Array containing the time when the packets are correctly
received.
```

```
        startRX[packet_id]=start_time
    }

    if (packet_type == "DATA" && event_type == "TX" && source == "<6" &&
packet_length == "570")
    {
        transmitted++
        # startTX: Array containing the start time of the transmissions.
       startTX[packet_id]=start_time
    }
}

END{
      # Loss
      if(transmitted>0) {
            loss=(transmitted-received)/transmitted
      } else {
            loss=0
      }
    # Delay: We check every row of the matrix and compute the delay between
the start of the transmission and when the packet is received.
    for (id in startRX)
    {
          if (id in startTX)
       {
              start = startTX[id]
           end = startRX[id]
           duration=end-start
           total=total+duration
           tx++
            }
      }
      if(tx>0) {
            delay=(total/tx)/tics_per_second
      } else {
            delay=0
      }
    # Throughput [bps]
    throughput=((8*size*received)/(simulation_time))

    # Formatting the output.
    printf("Packets transmitted: %d;\n", transmitted) >> output
    printf("Packets received: %d;\n", received) >> output
    printf("Losses: %f;\n", loss) >> output
    printf("Throughput: %f;\n", throughput) >> output
    printf("Delay: %f;\n", delay) >> output

    close(output)
}
```

The throughput is calculated dividing the received packets by the time of simulation.

The delay is calculated identifying the pairs of TX and RX, then calculating the delay of each individual transmission and getting the average.

The filter is invoked with the following command.

```
awk -f filter.awk < results.tr
```

Here is an example of the output of the filter:

```
$ cat 20_100_7.tr | awk -f ~/filtro.awk
Packets transmitted: 6282;
Packets received: 5208;
Losses: 0.170965;
Throughput: 260400.000000;
Delay: 0.179080;
```

This explanation has shown the usage of the different filters. The rest of the filters developed can be found in the annexes.

6.2 Description of the scenario

The considered scenario consists of a VANET deployed over a highway. Two base stations, one on each side of the road, are transmitting information to one of the nodes. This node can be out of range, so the VANET has to rely on multi-hop to deliver the packages. In particular, the routing protocol used is AODV.

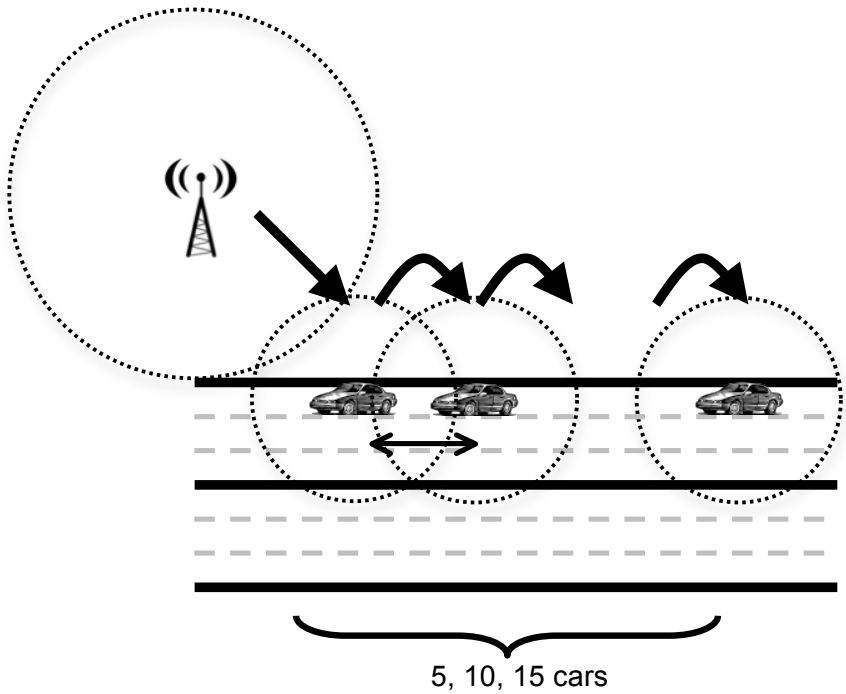


Fig. 6.14: Schema of the scenario.

Fig. 6.14 summarizes the basic physical parameters of the scenario. The road features 3 lanes in each way. The lanes are 20 meters wide due to a simulator problem that will be explained in the 6.4 section.

The scenario was simulated under different levels of congestion (10, 20 and 30 cars). The maximum speed of the nodes is also a parameter and it was set to 80,100 and 120 km/h. These speeds are the ones covered by the law in this kind of roads on Spain.

Regarding the simulation time, it was set to 80 seconds. This time was enough to ensure that each group of cars were in range of both base stations through the simulation.

The size of the transmitted packets is 500 bytes and is fixed. When sent, there's an overhead of 70 bytes. This packet size was chosen due to the conclusions of [52].

Another variable parameter is the transmission range. The simulator has an option to specify the "Desired transmission range" determined by the distance radius to cover. In particular two values were used R=75m and R=100m.

Regarding the physical layer, although one of the project goals was using IEEE 802.11p, some stability problems explained in section 6.4 were the reason why 802.11b was used in the base station and cars.

To analyze the performance of the scenario, we have carried out more than 300 simulations of data transmission between the nodes. Using the simulator NCTUns 6.0, we developed simulations for all the combinations of the following parameters. Table 6.1 summarizes all the parameters commented.

Table 6.1: Parameters used in the simulations.

Parameter	Values
Simultaneous transmissions	2
Max. Speed (km/h)	{80,100,120}
Number of nodes	{10,20,30}
Desired transmission range	{75,100}
Type of font	Constant bitrate
Transport protocol	UDP
Packet size	500 bytes
Bandwith	100 Mbps
Routing protocol	AODV
Physical layer	IEEE 802.11b

For every combination of these parameters, we performed between 10 and 20 simulations.

6.3 Simulation results

The analysis of the results can be divided in two parts. The first one focuses in the analyses of one particular simulated case, to help understand how the system works. On the other hand, the second part of the analysis studies the global metrics of all the cases and how the different parameters affect to the system performance.

6.3.1 Temporal analysis

The analysis of this section is base in a simulation with 20 cars moving at 100 km/h and a DTR of 75m. Both RSUs are transmitting a constant traffic of UDP packets to the last car of the row that is closest to them. At the beginning, the packets are routed through the first cars of each row, and then retransmitted sequentially until they arrive to the last car.

As the cars advance, these routes get recalculated until the last car of each row gets in range of his base station and the transmission is done directly. Then when the cars get out of range, transmissions stop until is possible to route packets using the cars coming in the opposite direction.

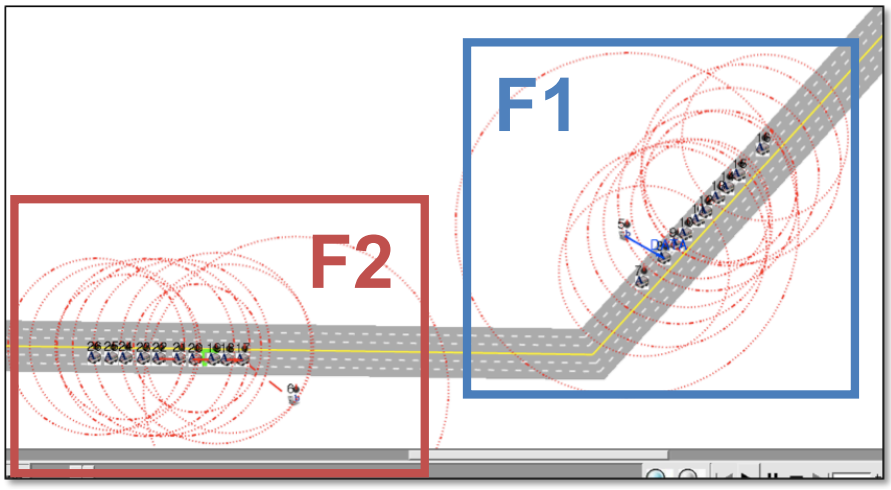


Fig. 6.15: Separation between cases F1 and F2.

In theory, both rows of cars should have a similar behaviour, but they don't. This is probably caused by the geometric differences. To analyze the differences, all the results obtained are separated for the two cases, using different filters for each one, namely F1 and F2. Fig. 6.15 represents this division.

The temporal evolution of the nodes can be seen in Fig. 6.16, Fig. 6.17 and Fig. 6.18.

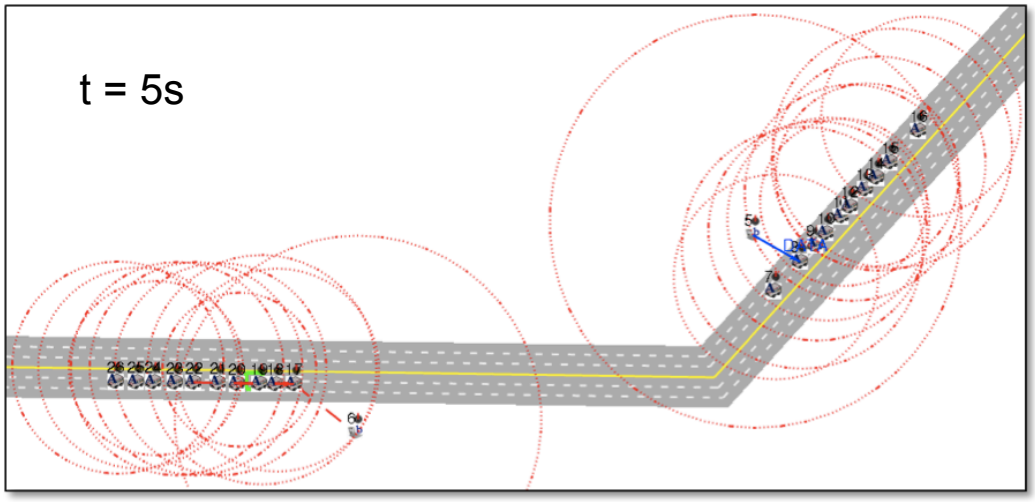


Fig. 6.16: Status of the scenario at time 5s.

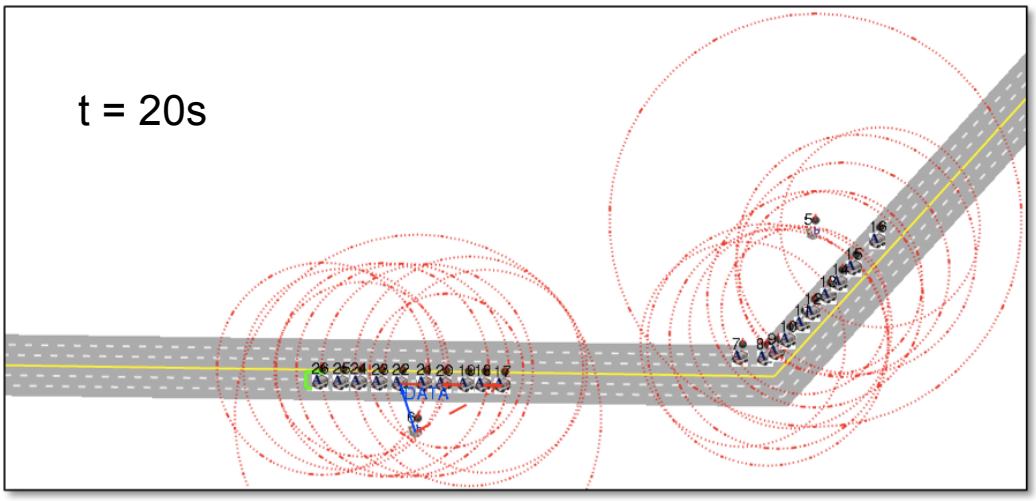


Fig. 6.17: Status of the scenario at time 20s.

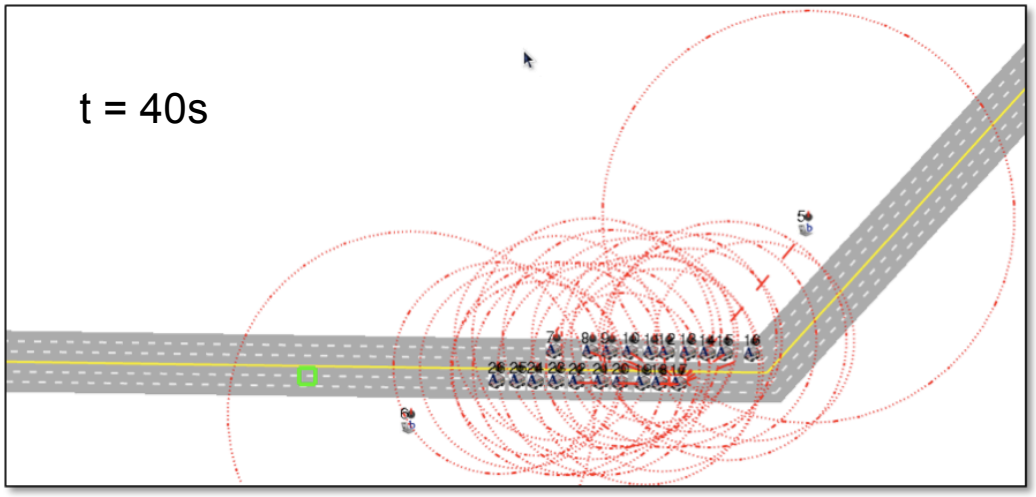


Fig. 6.18: Status of the scenario at time 40s.

Apparently, from 5 seconds on, both RSUs should be transmitting. Close to 20 seconds, in both cases the transmission should start to be direct instead of multihop. Then, from 40 seconds on, the group of cars fall out of the range of the RSUs and there should start to appear cases of “backwards” transmissions, or transmissions routed using the cars going in the opposite direction.

Fig. 6.19 and Fig. 6.20 show the number of packets transmitted and how many of them are received for F1 and F2 respectively. The results are displayed in intervals of 5 seconds, giving a idea of how the two cases work. Blue columns represent the number of transmitted packets and red columns, the received ones. This statistic refers to end-to-end transmissions.

For F1, the base station reaches the cars from the very beginning, but these early transmissions are dropped in a high percentage. In contrast, in the F2 case, the base station is not able to reach the cars until the 5 to 10 seconds interval. However, the cars receive these transmissions with fewer losses than for the F1 case.

From 40 seconds on, no more packets are transmitted, even for the case of 30 cars, so the simulation time of 80 seconds was excessive. This graphic was extracted from one

single sample of the simulations, in particular the case of 20 cars, 75 m range and 100 km/h.

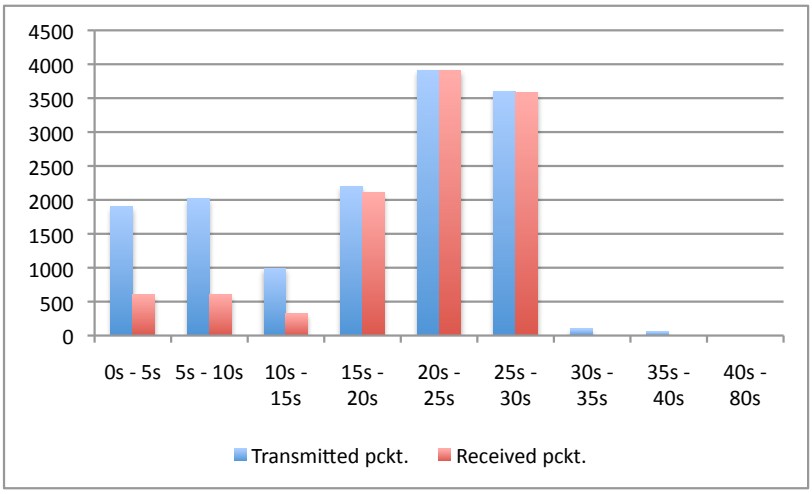


Fig. 6.19: Number of packets tx/rx for filter 1.

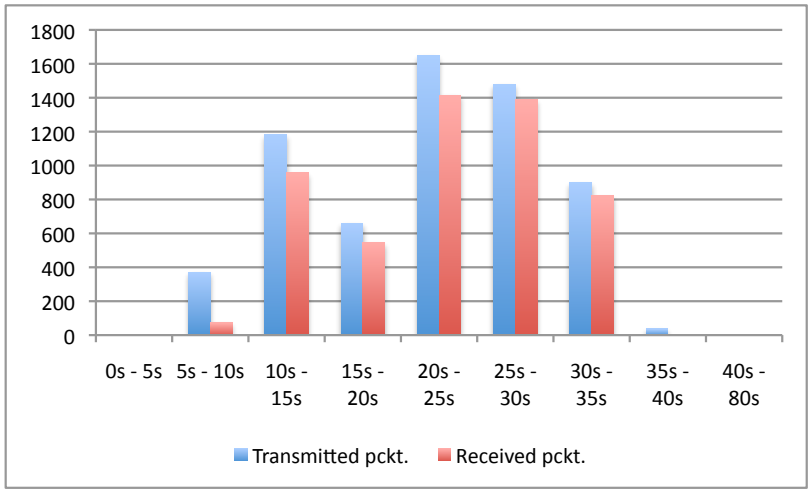


Fig. 6.20: Number of packets tx/rx for filter 2.

Fig. 6.21 shows the ratio of number of collisions per packets transmitted and retransmitted. The line in blue represents the statistic for the F1 case and the red one, for F2. Between 0 and 15 seconds this ratio is high, reaching the 25% in the F1 case. This high number of collisions explains why the number of received packets in this interval is so low.

However, from 15 seconds, this statistic falls drastically. This decrease coincides with the moment when the cars start to surpass the RSU. In this instant, AODV routes get recalculated and, judging by the number of collisions, are more optimal.

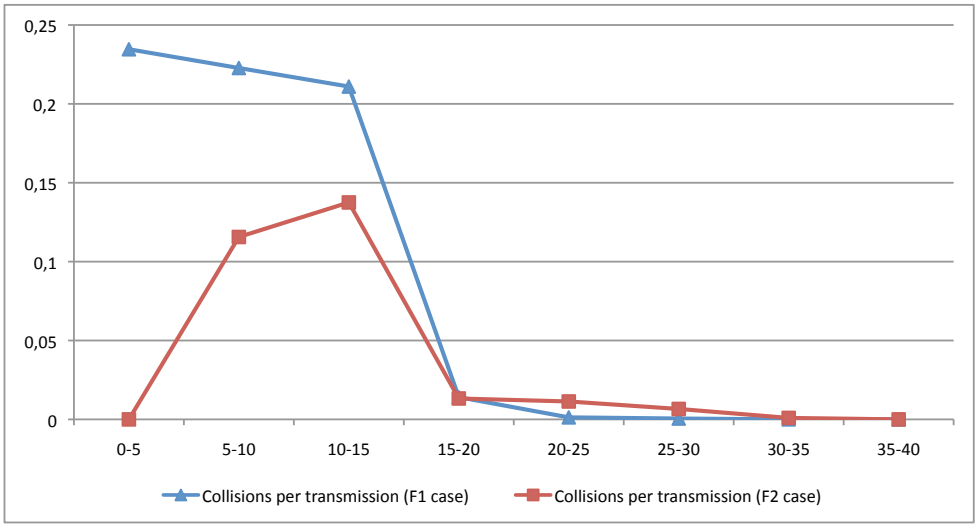


Fig. 6.21: Ratio number collisions per transmission.

Fig. 6.22 and Fig. 6.23 complement the data showed in Fig. 6.21 displaying the total number of collisions (blue columns), transmissions (red columns) and retransmissions (green columns) for both cases. Apart from the first 15 seconds, there are almost no collisions.

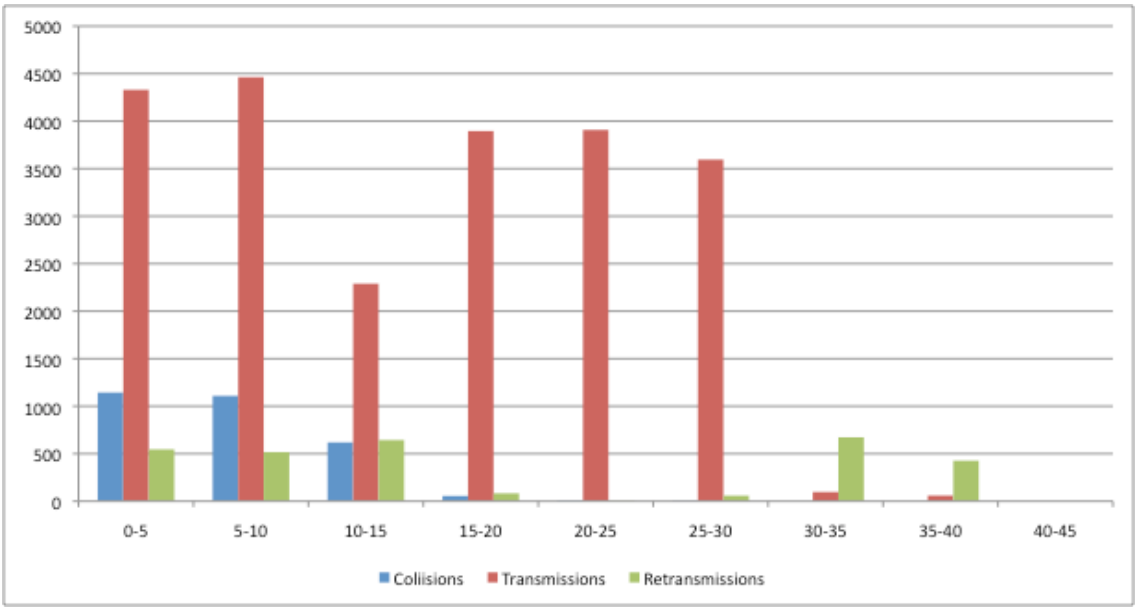


Fig. 6.22: Total collisions, transmissions and retransmissions for filter 1.

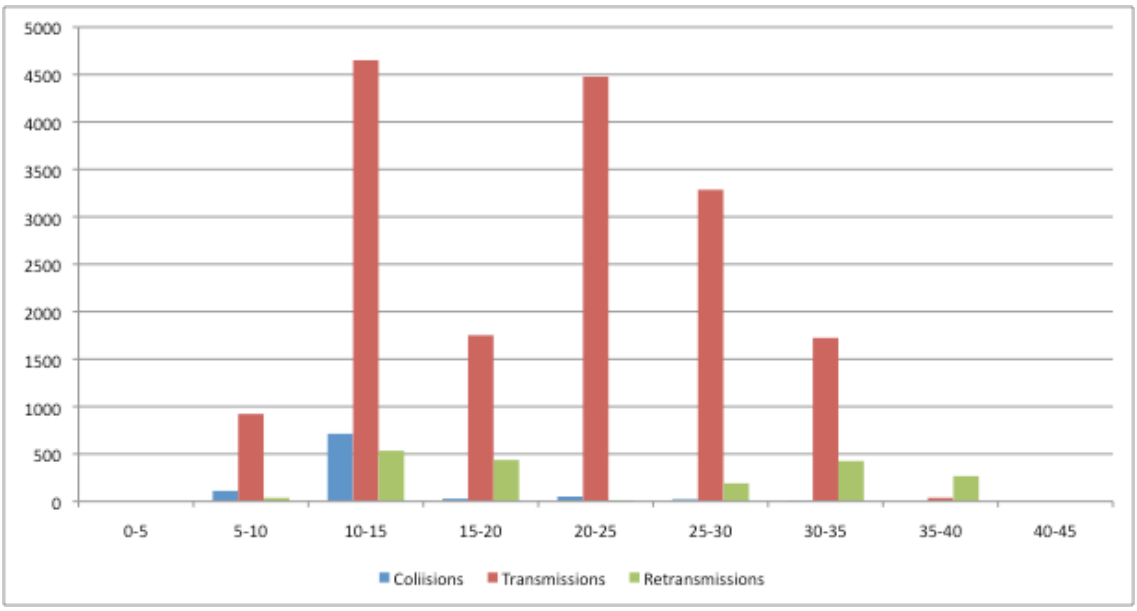
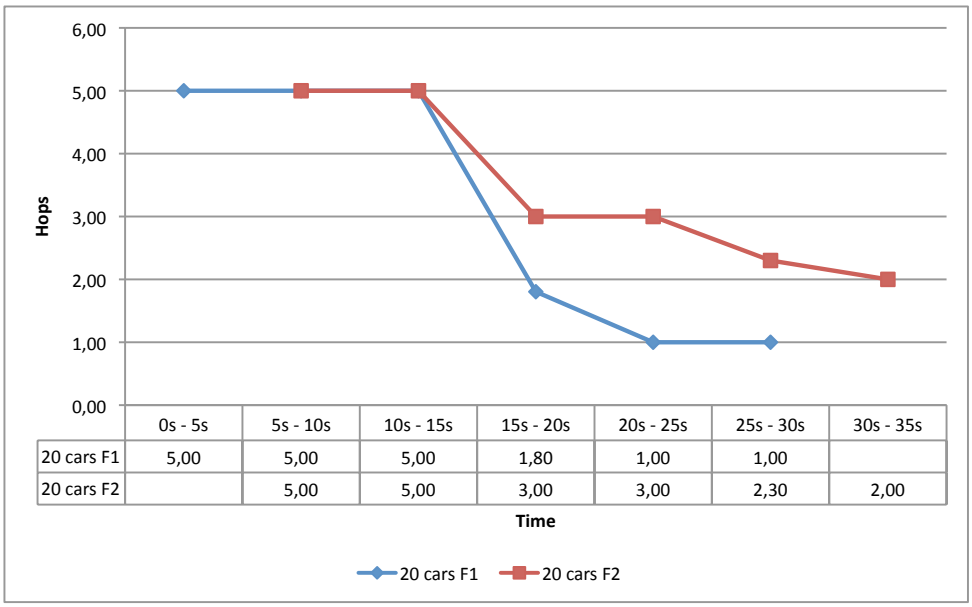


Fig. 6.23: Total collisions, transmissions and retransmissions for filter 2.

Fig. 6.24 shows the analysis of the average length in hops of the transmission routes. The results are displayed in intervals of 5 seconds, which allows understanding better how the simulator implementation of the AODV protocol works.

The “Active route timeout” parameter of the simulator defines the interval of time after which the routes are recalculated. In our simulations this parameter is set to 3 seconds.

For F2, in average, the routes are longer than for F1. In fact, the transmission in F2 is never done in one hop although it would be possible. This shows one of the deficiencies of the AODV implementation used in NCTUns.



	0s - 5s	5s - 10s	10s - 15s	15s - 20s	20s - 25s	25s - 30s	30s - 35s
20 cars F1	5,00	5,00	5,00	1,80	1,00	1,00	
20 cars F2		5,00	5,00	3,00	3,00	2,30	2,00

Fig. 6.24: Average length of routes through time.

On the other hand, Table 6.2 summarizes the routes used in every 5 seconds interval. At first sight surprises that in the interval between 25 and 30 seconds there are 5 different routes. Considering that the routes are getting recalculated every 3 seconds,

there should be an additional mechanism in the AODV implementation of NCTUns that triggers route recalculation.

In this case the base station is not using the cars coming in the opposite direction to route the packets, however a similar study performed on another case, with more cars and higher range of transmission shows that this “backwards” transmission actually takes place.

Table 6.2: Routes used through simulation time.

Interval	Routes used in F1	Routes used in F2
0s – 5s	5-8-11-12-14-16	-
5s – 10s	5-8-11-12-14-16	6-17-19-21-23-26
10s – 15s	5-8-11-12-14-16	6-17-19-21-23-26
15s – 20s	5-14-16 5-15-16 5-16	6-22-23-26 6-22-24-26
20s – 25s	5-16	6-22-24-26
25s – 30s	5-16	6-22-21-23-26 6-22-23-26 6-22-24-26 6-24-26 6-25-26
30s – 35s	-	6-25-26

Fig. 6.25 and Fig. 6.26 show the evolution through the simulation time of the three metrics studied in the next section, namely loss ratio (blue line), end-to-end delay (green line) and throughput (red line). These metrics support what has been commented before. For example, the loss ratios in the first 15 seconds are high, reaching levels of almost 80%.

Also, in average, the delays for the F1 case are lower than for the F2. This is related with the F1 routes being shorter in average than the F2 ones.

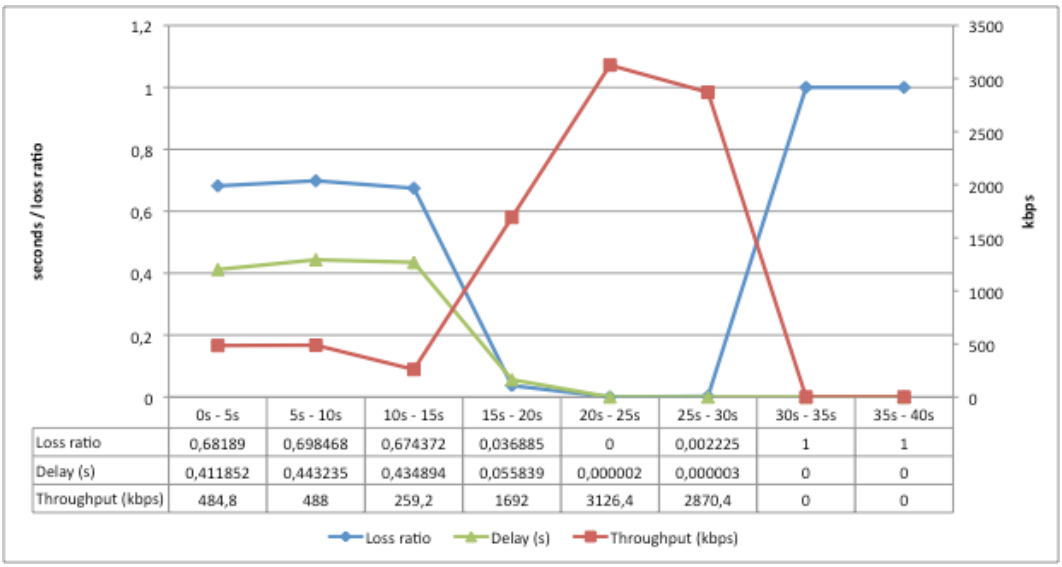
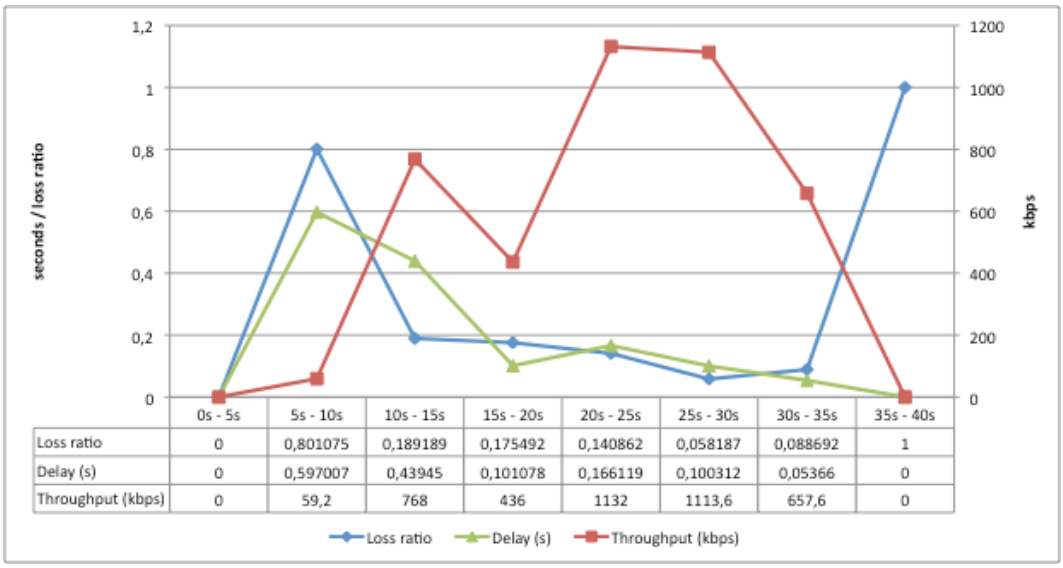


Fig. 6.25: Metrics through simulation time for filter 1.



	0s - 5s	5s - 10s	10s - 15s	15s - 20s	20s - 25s	25s - 30s	30s - 35s	35s - 40s
Loss ratio	0	0,801075	0,189189	0,175492	0,140862	0,058187	0,088692	1
Delay (s)	0	0,597007	0,43945	0,101078	0,166119	0,100312	0,05366	0
Throughput (kbps)	0	59,2	768	436	1132	1113,6	657,6	0

Fig. 6.26: Metrics through simulation time for filter 2.

6.3.2 Global analysis of the metrics

This second part of the study focused on the study of three metrics: the loss ratio, the end-to-end delay and the throughput. Additional metrics are included to support the thesis, like the length of the routes calculated by the AODV protocol or the total number of collisions.

The metrics of all the figures in this section have been calculated averaging the results of at least 10 simulation samples. In addition, every value has its own thrust interval of 99%, which helps to understand how disperse are some values.

6.3.2.1 Study of packet losses

Fig. 6.27 and Fig. 6.28 show the ratio of losses for each number of nodes. As most graphs on this section, every group of 6 columns represent the values for one amount of cars. The first group show the results for 10 cars, the second, for 20 cars, and 30 cars for the third group. In every one of these groups, the first three columns represent the results for the F1 case and the rest, for the F2 case. Each value is represented with its thrust interval of 99%.

As the first group of columns show, for 10 cars, we have exceptionally low losses, and as the number of cars increases, we have higher losses. In general, the losses increase with the number of cars, but never surpass 45%. This behaviour is consistent, and is caused because as the number of nodes increases, the transmissions are performed through more hops, and additional interchanges mean more probability of transmission failure. Moreover, this also means a higher congestion, which produces a higher collision rate.

If we focus on speed, in most cases we have higher or equal losses with higher speeds. The reason is that vehicles behind another vehicle in the VANET scenario must adapt their speed to the speed of the car in front of them, according to the CarAgent mobility model implemented in NCTUns. That is, the first car can go faster than the others behind, so that when the link to the second car behind breaks, an alternative route must be found. This is especially noticeable for high speeds, whereas

for low speeds cars tend to remain in groups so that the links last longer, which produces lower losses.

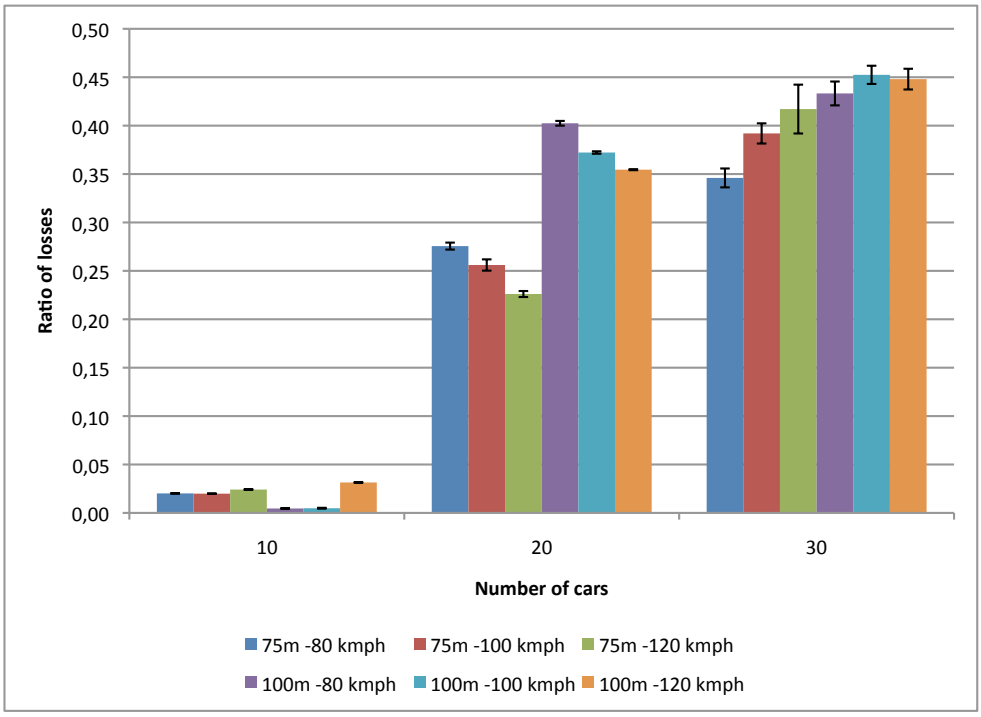


Fig. 6.27: Average ratio of packet losses and 99% thrust intervals for the F1 case.

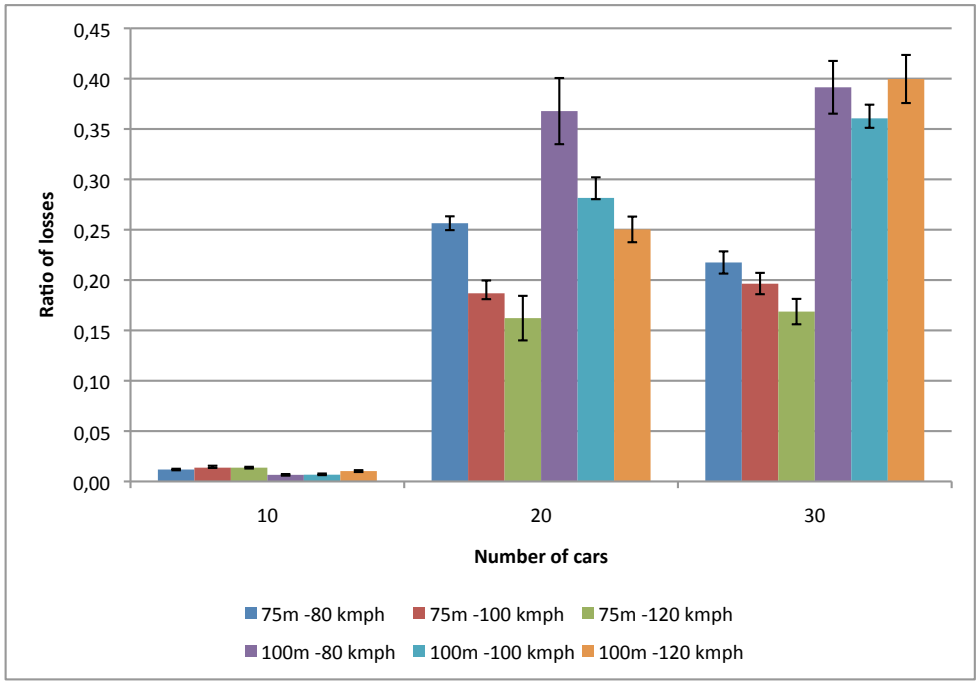


Fig. 6.28: Average ratio of packet losses and 99% thrust intervals for the F2 case.

Another consistent behaviour is that, as transmission range increases, the losses are higher. The reason for that is the higher level of congestion that appears when the range of transmission increases.

Loss values are higher for the case of F1 than for the ones of F2, because in average, the computed routes in for F2 have more hops than for F1. However the values of filter 2 are more disperse. This second behaviour has no obvious reasons, since both cases are symmetric.

Table 6.3: Average length of routes.

Case		Average number of hops
20 cars/100 kmph/75 m	**Filter 1**	1.70
	Filter 2	3.00
20 cars/100 kmph/100 m	**Filter 1**	1.06
	Filter 2	1.85
30 cars/100 kmph/75 m	**Filter 1**	1.85
	Filter 2	2.06
30 cars/100 kmph/100 m	**Filter 1**	1.18
	Filter 2	1.58

Table 6.3 displays the average length (in hops) of the calculated routes in multiple cases. The values for the F2 case are consistently higher than fro the F1 case. Also, when focusing on the number of cars, in average, the routes are shorter for the cases with 30 cars than for the cases with 20 cars.

Regarding the transmission range, the values show that, if transmission range increases, the routes have less hops, which is logical.

6.3.2.2 Study of end-to-end delays

Fig. 6.27 and Fig. 6.28 show the average end-to-end delays for each case. As it happened before with loss ratios, delays in the case of 10 cars the delays are extremely low. Then, for 20 and 30 cars, the delays are much higher, although they never surpass 230 ms (Case with 20 cars, 80 km/h and 75m DTR).

In this case, having more cars does not translate into higher delays. In fact, for 30 cars, the delays are similar or even lower than for 20 cars. This makes sense, because as Table 1 showed, the average number of hops with 20 cars is higher than with 30 cars. This is also the reason why delay values are generally higher for F2 than for F1.

Regarding speed, the trend is that as speed increases, the end-to-end delays decrease. However, in most of the cases this decrease is not very accused, and it can be said that delays are fairly stable with node speed.

The improvement achieved increasing the transmission range is significant, specially in the case of 20 cars. The reason is that a higher range of transmission allow shorter routes (as shown on Table 3), and therefore the transmission is faster.

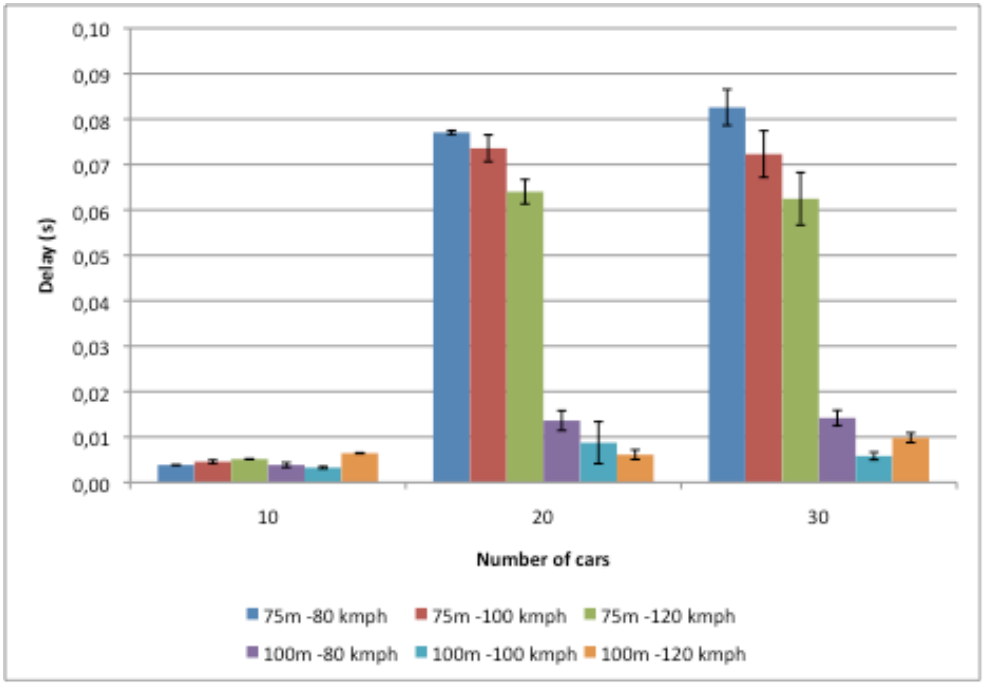


Fig. 6.29: Average end-to-end delays and 99% thrust intervals for the F1 case.

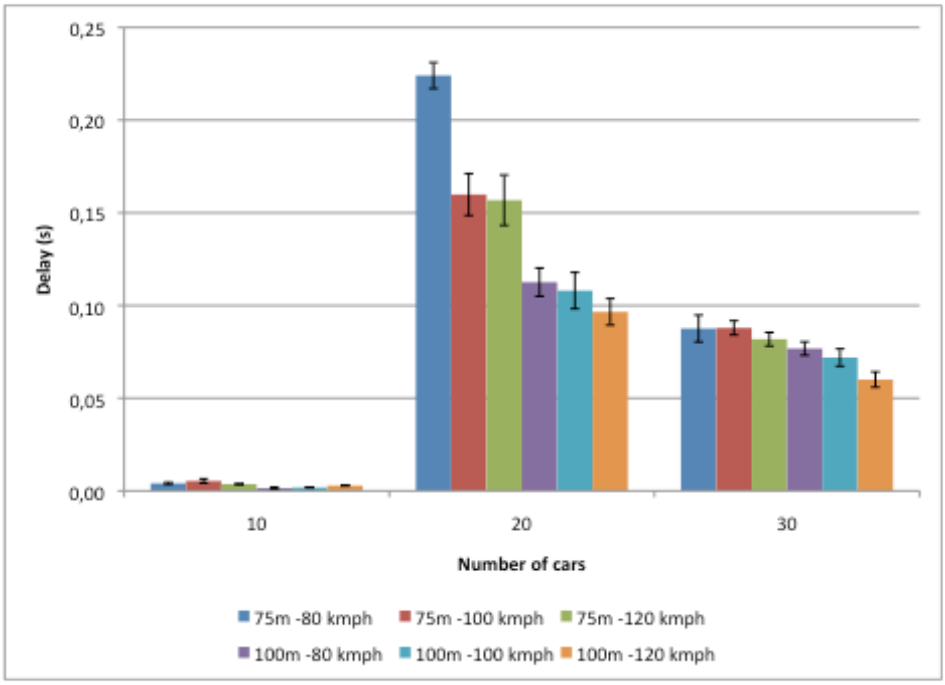


Fig. 6.30: Average end-to-end delays and 99% thrust intervals for the F2 case.

Table 6.4 shows the total number of packets transmitted, collisions and the collision per transmission ratio for different cases. Congestion, as total number of packets transmitted, is higher as the number of nodes increase. Logically, this generates a higher number of collisions.

Something surprising is that higher transmission ranges decrease the number of collisions, even though the number of transmitted packets is higher.

Table 6.4: Total packets transmitted, collisions and collision per transmission ratio.

Number of cars	Range	Total TX	Collisions	Ratio coll/tx
20	75	44347	3905	0,088
20	100	52404	2502	0,048
30	75	84582	9659	0,114
30	100	108693	6218	0,057

6.3.2.3 Study of throughput

Fig. 6.31 and Fig. 6.32 display the throughput values obtained. The maximum throughput reached is 1.2 Mbps wich is very low compared with the font throughput of 100 Mbps, but it is important to remark again how the hroughput is being calculated. This figures are the result of dividing the number of bits received by the entire time of simulation, instead of the time of transmission.

Throughput values are much higher with 10 cars in the scenario than for the other caes. That was predictable, since this is the case with lowest delays and loss ratios. Throughput values extracted from the simulations are very stable, with low trust intervals. Something noticeable is that throughput values remain stable as speed increases.

Regarding the range of transmission, a higher range produces a drastic increase in the throughput, this is related to the shorter lenth of the routes when the transmission range are higher (as shown on Table 6.3).

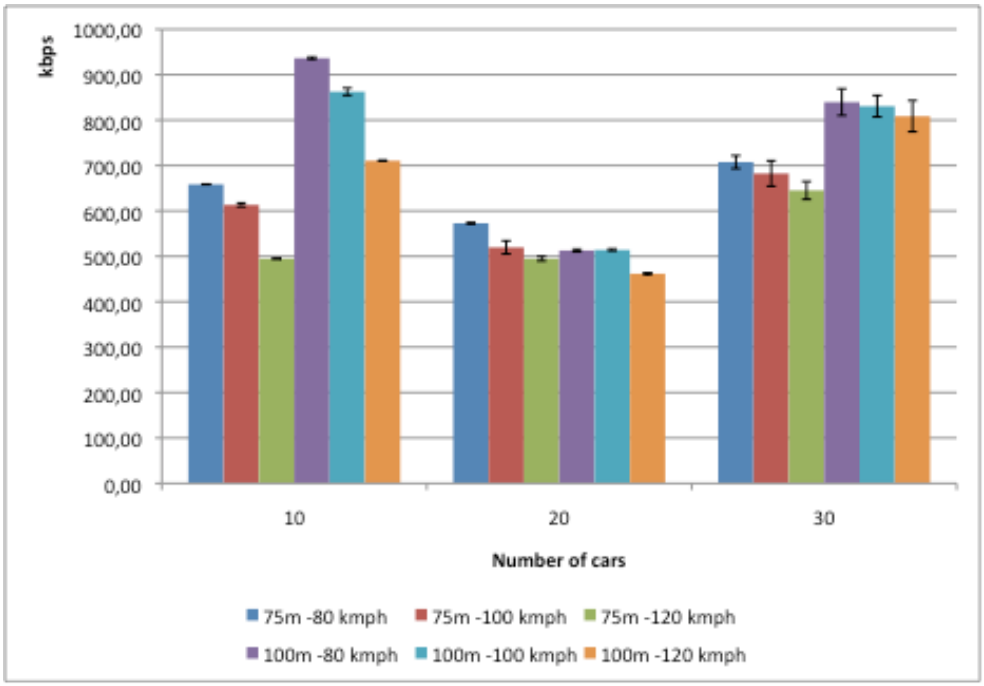


Fig. 6.31: Average throughput values and 99% thrust intervals for the F1 case.

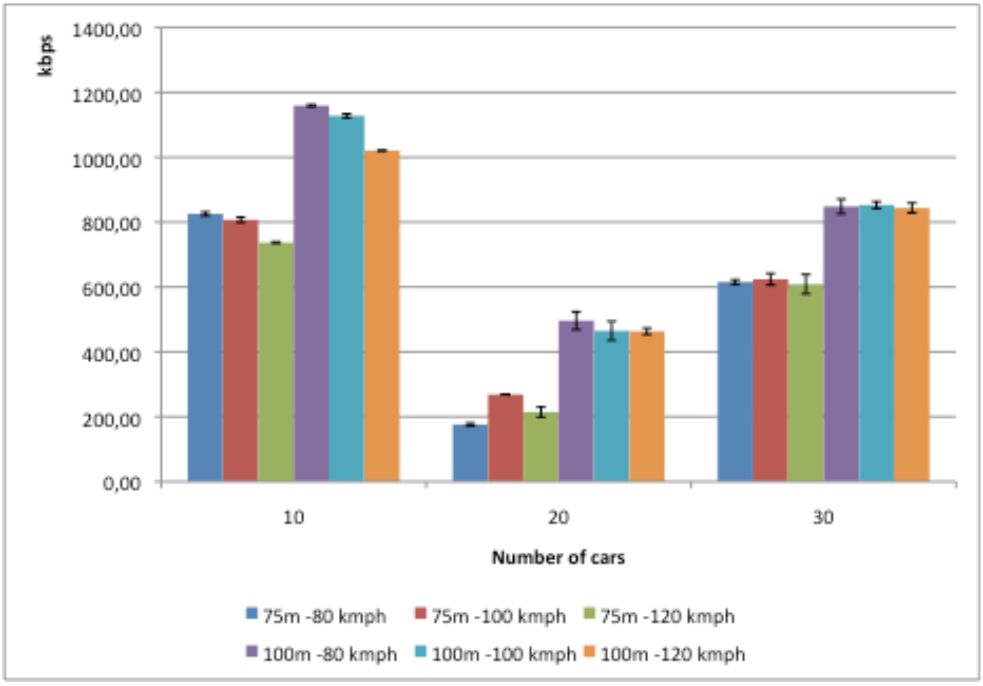


Fig. 6.32: Average throughput values and 99% thrust intervals for the F2 case.

6.4 Problems found during simulation

NCTUns is still under development, and therefore, it still has bugs and the performance is not optimal. Some of the problems found in the simulation process are described below.

The most important setback is that 802.11p protocol implementation is not yet complete. There are a lot of stub functions (eg: transmit power cannot be set because some of the routines are not implemented). Therefore, to work with this protocol it is necessary to write your own code to integrate the original one. For instance, NCTUns simulator does not support multihop function for the 802.11p network so 802.11b protocol was chosen in order to perform the tests.

Simulations of a 802.11b scenario are blocked when using DSR routing protocol. The performance of NCTUns simulator with DSR is not stable: after testing scenarios for a wide range of speeds only worked for 55 Km/h one time and was blocked for next simulations. This problem is also documented in [52].

Other problems are related only to the graphical interface. For example, after simulating a scenario, the action of modifying the position of cars or base stations makes the next simulation being blocked. It is necessary to kill some car agent processes on the command prompt in order to simulate again.

Another of these graphical interface errors happens when configuring a scenario. The problem consists in that transmission and interference range values for both cars and base stations are not saved from one simulation to another. This causes that each simulation has to be configured from the beginning and makes difficult to launch simulations in batches.

One last GUI-related problem encountered was that the road designer does not work properly if the lanes are not at least 20 meters wide. If the lanes are narrower, the intersections are not well interpreted. One related problem is the impossibility to import

external maps properly since roundabouts and intersections are not correctly interpreted.

6.5 Conclusions

The first part of the study focused in the analysis of the evolution of one single case through the simulation time. The analysis showed that the first routes calculated by the routing protocol AODV are far from optimal, and until the cars get very close to the base station, the losses are very high.

The routing protocol calculates periodically new routes but it has a timeout. During the time in which the routes are maintained, AODV tends to maintain suboptimal routes.

The second part of the study focused on the analysis of three metrics: the loss ratio, the end-to-end delay and the throughput. Other metrics such as the length of the routes or the total number of transmissions, retransmissions and collisions have been used to support the reasoning.

The study showed that as the number of cars in the scenario increases, the losses increase. This was predictable, and it is the result of a higher congestion. However, a higher number of nodes gives more flexibility to the routing protocol when it computes the routes and so the connexions are more difficult to break.

Focusing on speed, higher speeds make the loss ratio decrease. This is caused because when working with higher speeds the cars tend to remain in groups and the links last longer. Also, as the cars are closer, the delays are lower. This behaviour is more of an effect caused by the particularities of the simulator mobility generator than a general trend.

Regarding the transmission range, it can be concluded that a higher transmission range improves the performance of the system. Although an increment of the range causes the loss ratio to increase, the delays and the throughput are generally better.

7. Conclusions and future work

The main motivation of this project was to study and simulate a VANET (Vehicular Ad-hoc Network) scenario. In chapter 2, the concept of VANET was introduced as well as some of its possible applications. Although most of them focus on road safety, VANETs also open the door to entertainment applications on vehicles.

This kind of networks has been the subject of study of multiple projects around the world, and the most important ones were also mentioned in this chapter. Of all this initiatives, the North-American WAVE initiative deserves a special mention, since it is developing communications standards over which other research groups can run their applications.

In the third chapter the concept of WSN (Wireless Sensor Network) was introduced. This kind of networks is formed between sensors distributed in an area that gather information. When paired with a VANET forming a HSVN (Hybrid sensor Vehicular Network) this kind of networks open a lot of possibilities for new applications.

The fourth chapter focused in the study of the different routing protocols available for VANETs and WSNs. Since VANETs present several particular properties, they need routing protocols able to solve their specific problems. Some of the problems that these algorithms must fight are high mobility and the short duration of routes. That is why some routing protocols study the use complementary information such as car position, trajectories or speeds gathered via GPS.

Although there are a lot of routing protocols, not many of them are implemented in network simulators. Particularly, in the simulations carried out on this project, the routing protocol chosen was AODV (Ad-hoc On Demand Distance Vector).

Chapter 5 focused in VANET simulators. In chapter 5 the different simulation tools available for VANET simulation. Of all the simulators, the one used in this project was NCTUns. The main reasons were that it was a free of use open-source simulator and that it had important capabilities for wireless vehicular network research. However, other simulators were also tested prior to this decision such as Vanet Mobisim, but they were finally dismissed.

Chapter 6 described the simulation process of a VANET highway scenario. In particular, the performed simulations have used interfaces using IEEE 802.11b. This was not the preferred option, since NCTUns supposedly supported the new standard IEEE 802.11p, but working with this protocol showed that the implementation of this protocol had still some bugs that made impossible simulation scenarios stably.

Knowing from the start that this physical layer was not the optimal selection, simulation results have confirmed that although IEEE 802.11b is a viable solution for low congestion environments, when the number of cars in the scenario and their speed increase the performance is much poorer.

The simulation scenarios have been tested against a simple flooding algorithm in order to evaluate their efficiency. Developing and running real applications over the network, would constitute one possible future research direction.

Apart from testing if the VANET worked, the simulation has been repeated changing some parameters to observe how this changes affected the performance of the system. In particular, these parameters have been the number of nodes in the network, their speed and the transmission power of the base station, represented in this case by the parameter “Desired Transmission Range”.

Results have shown that, as the number of cars increases there are more collisions and therefore, more packet losses. However, a higher number of nodes allows the routing protocol to find new routes and therefore, there is a higher probability that the transmission ends successfully.

The same happens with the node’s speed. The routes last longer unbroken at higher speeds. However, this is a particularity of the NCTUns simulator caused by its mobility generator, and must not be taken as a general result.

The parameter that has revealed key in the operation of the VANET is transmission range. In general, a higher transmission range makes the VANET work more efficiently, since the routes calculated by AODV are shorter.

The difficulties during the simulation have been remarkable, since NCTUns is still in development. At the time of publication of this work, the NCTUns developer team has just announced that the simulator is going commercial under the name of “EstiNet”. Hopefully, new releases of the simulator will address the bugs that made impossible simulating scenarios using IEEE 802.11p.

During the development of this project some ideas surged to understand better and improve the obtained results. Some of these ideas have been used, but due to the limited duration of this project it has not been possible to incorporate all of them.

Some of these ideas can constitute future research lines:

- Repeat the simulations using 802.11p to measure how much does the performance improves.
- Focus on the tuning of the routing protocol. This covers adjusting the parameters of the AODV protocol to improve the performance but also using other more specific routing protocols for VANETs, e.g. GSR (Geographic Source Routing) [56], SAR (Spatial Aware Routing) [57], and VADD (Vehicular Assisted Data Delivery) [58].
- The tests have been performed using a constant bitrate font. It will be interesting to test the system with an external application to test services like

local awareness. This would suppose developing and integrating this application with the simulator.

- The network designed in this project consists in a VANET. A more complex system would be and Hybrid Sensor Vehicular Network (HSVN). This kind of system will include a network of sensors distributed along the road that will also communicate data to the vehicles.

8. Annexes

8.1 Types of sensors

All the applications for Wireless Sensor Networks have no limit in industries and deployments having specific technology requirements such as reliability, battery-life, range, frequencies, topologies, size of the network, sampling rate and sensor use.

To address the unique requirements of individual applications, Crossbow [CRO09a] provides a broad portfolio of wireless sensor network products that allow you to choose the optimal solution for one's industry, application and geographical requirements.

A variety of Development Kits are designed to provide customers with all the tools needed to evaluate, design and develop a WSN.

Wireless Modules are available in many different forms [CRO09a]. Depending on the radio frequency and platform requirements you may have for your application, there is a variety of wireless modules to choose from whether it is our popular IRIS or TelosB Mote to our high-power Imote2 device.

Table 8.1: Types of sensors.

Sensor	Product Features
IRIS OEM Module 2.4GHz 	• 2,4 GHz IEEE 802.15.4, Tiny Wireless Measurement System • Designed Specifically for Deeply Embedded Sensor Networks • 250 kbps, High Data Rate Radio • Wireless Communications with Every Node as Router Capability • Expansion Connector for Light, Temperature, RH, Barometric Pressure, Acceleration/Seismic, Acoustic, Magnetic and other Crossbow Sensor Board
IRIS 2.4GHz	• 2.4 GHz IEEE 802.15.4, Tiny Wireless Measurement

Sensor	Product Features
	System • Up to Three Times Improved Radio Range and Twice the Program Memory Over Previous MICA Motes • Designed Specifically for Deeply Embedded Sensor Networks • 250 kbps, High Data Rate Radio • Wireless Communications with Every Node as Router Capability
MICAz 2.4GHz 	• Wireless Platform for Low-Power Sensor Networks • 2.4 GHz, IEEE 802.15.4 compliant • FCC Certified • 250 kbps, High Data Rate Radio • Multi Year Battery Life • Designed Specifically for Deeply Embedded Sensor Networks • Wireless Communications with Every Node as Router Capability
MICA2 868, 916 MHz 	• Wireless Platform for Low-Power Sensor Networks • 868/916 MHz Multi-Channel Radio Transceiver • 38.4 kbps Data Rate Radio • Multi-Year Battery-Life • Designed Specifically for Deeply Embedded Sensor Networks • Wireless Communications with Every Node as Router Capability
Imote2.NET Edition 	• .NET Micro Framework - Pre-installed from the Factory • PXA271 XScale® Processor at 13–416MHz • Wireless MMX DSP Coprocessor • 256kB SRAM, 32MB FLASH, 32MB SDRAM • Integrated 802.15.4 Radio • Multi-color Status Indicator LED • Integrated 2.4GHz Antenna • USB Client With On board mini-B • Application Specific I/O: I2S, AC97, Camera Chip Interface, JTAG

Sensor	Product Features
Imote2 	• PXA271 XScale® Processor at 13–416MHz • Wireless MMX DSP Coprocessor • 256kB SRAM, 32MB FLASH, 32MB SDRAM • Integrated 802.15.4 Radio • Multi-color Status Indicator LED • Integrated 2.4GHz Antenna • USB Client With On-board mini-B • Application Specific I/O: I2S, AC97, Camera Chip Interface, JTAG
TelosB 	• IEEE 802.15.4 compliant • 250 kbps, high data rate radio • TI MSP430 microcontroller with 10kB RAM • Integrated onboard antenna • Data collection and programming via USB interface • Open-source operating system • Optional integrated temperature, light and humidity sensor
Cricket 	• High Performance MICA2 Wireless Location System • Ultrasound Transmitter and Receiver for Time of Flight Ranging • Centimeter Level Accuracy/ Resolution with Decentralized and Scalable Operation • Embedded or External Antenna Option

8.2 Filters used in the analysis

8.2.1 Route filter

This filter calculates the routes that followed the packets that are correctly received. The code of the filter is shown below.

```
BEGIN{
    # output: file or device where the output results will be redirected.
    output="/dev/stdout"
    # size: Size of packet in bytes.
    size=500
    # tics_per_second: Number of tics are in one second. For our simulation,
1tic = 100ns
    tics_per_second=1e7
    # simulation_time: Length of the simulation.
    simulation_time=80
    # received: Count of the received packets that follow the restrictions.
    received=0
    # transmitted: Count of the transmitted packets that follow the
restrictions.
    transmitted=0
    # total: Summatory of the delay times.
    total=0
    # tx: Count of packets transmitted and received.
    tx=0
    start=start*tics_per_second
    end=end*tics_per_second
}

{
    # Mapping of the trace parameters to variables.
    # <protocol> <event type> <time> <duration> <packet type>
<source/destination>
    #<tx rx> <packet ID> <packet length> <count of re-tx> <drop reason>
<freq. channel>
    # event: Event type.
    event_type=$2
    # packet_type: Packet type.
    packet_type=$5
    # source: Node ID that starts the transmission.
    source=substr($8,2,length($8))
    middle=$9
    # destination: Node ID to whom the packets are addressed.
    destination=substr($10,1,length($10)-1)
    # start_time: Time (in tics) when the transmission is started.
    start_time=$3
    # packet_id: Packet's unique id.
    packet_id=$11
    # packet_length: Length of the packet (counting the headers).
    packet_length=$12
```

```
    if (packet_type == "DATA" && event_type == "TX" && packet_length ==
"570") {
      if(source == "5") {
            if ( !( packet_id in F1 ) ) {
                        F1[packet_id] = start_time
            }
      }
        if(source == "6") {
            if ( !( packet_id in F2 ) ) {
                        F2[packet_id] = start_time
            }
      }
    }

    node_repeated = 0

    if (packet_type == "DATA" && event_type == "RX" && packet_length ==
"570")
    {
        if (packet_id in routes) {
            split(routes[packet_id],nodes,",")
            for ( i in nodes ) {
                if ( nodes[i] == middle ) {
                    node_repeated = 1
                }
            }
            if ( node_repeated == 0 ) {
                routes[packet_id]=routes[packet_id] "," middle
            }
        } else {
            if( (packet_id in F1) || (packet_id in F2) ) {
                        routes[packet_id] = source "," middle
            }
             }
    }

}

END{

    #if ( filter == "F1" ) {
        for ( id in F1 ) {
            if ( id in routes ) {
                if ( F1[id] > start && F1[id] < end ) {
                    last = substr( routes[id], length(routes[id])-1,
length(routes[id]) )
                    if (  last == "16" || last == "36" || last == "26" ||
last == "31" ) {
                        printf("F1 %d %d %s\n", F1[id], id, routes[id]) >
output
                    }
                }
            }
          }
    #}

    #if ( filter == "F2" ) {
        for ( id in F2 ) {
            if ( id in routes ) {
```

```
                if ( F2[id] > start && F2[id] < end ) {
                    last = substr( routes[id], length(routes[id])-1,
length(routes[id]) )
                    if (  last == "16" || last == "36" || last == "26" ||
last == "31" ) {
                        printf("F2 %d %d %s\n", F2[id], id, routes[id]) >
output
                    }
                }
            }
        }
    #}

    close(output)
}
```

The filter is invoked with the following command.

```
awk -f route_filter.awk -v start=<START TIME> -v end=<END TIME> < results.tr
```

Two parameters, START TIME and END TIME, are passed to the filter to define the interval of the simulation to analyze.

```
<GROUP OF CARS> <START TIME> <PACKET ID> <ROUTE>
```

Here is an example of the output of the filter.

```
[nctuns@fedora filtros]$ awk -f route_filter.awk -v start=0 -v end=80 <
results.tr | head -n 11000 | tail -n 5
F2 248327812 281113 6,25,26
F2 64043479 143194 6,17,18,20,23,26
F2 339910736 338662 6,26
F2 64065979 143195 6,17,18,20,23,26
F2 64075189 143196 6,17,18,20,23,26
```

8.2.2 Collision filter

This filter counts the number of collisions, transmissions and retransmissions. The code of the filter is shown below:

```
BEGIN{
    # output: file or device where the output results will be redirected.
    output="/dev/stdout"
    # size: Size of packet in bytes.
    size=500
    # tics_per_second: Number of tics are in one second. For our simulation,
1tic = 100ns
    tics_per_second=1e7
    # simulation_time: Length of the simulation.
    simulation_time=800
    # total: Summatory of the delay times.
    total=0
    # tx: Count of packets transmitted and received.
    tx=0
    collisions_f1=0
    collisions_f2=0
```

```
    tx_f1=0
    tx_f2=0
    rtx_f1=0
    rtx_f2=0
    interval=5
    interval_size=1
    raros=0
    print "INTERVAL;COLL_F1;COLL_F2;TX_F1;TX_F2;RTX_F1;RTX_F2;RAROS;"
    nodes["5"]="F1"
    nodes["6"]="F2"
    nodes["7"]="F1"
    nodes["8"]="F1"
    nodes["9"]="F1"
    nodes["10"]="F1"
    nodes["11"]="F1"
    nodes["12"]="F1"
    nodes["13"]="F1"
    nodes["14"]="F1"
    nodes["15"]="F1"
    nodes["16"]="F1"
    nodes["17"]="F2"
    nodes["18"]="F2"
    nodes["19"]="F2"
    nodes["20"]="F2"
    nodes["21"]="F2"
    nodes["22"]="F2"
    nodes["23"]="F2"
    nodes["24"]="F2"
    nodes["25"]="F2"
    nodes["26"]="F2"
}

{
      # Mapping of the trace parameters to variables.
      # <protocol> <event type> <time> <duration> <packet type>
<source/destination> <tx rx> <packet ID> <packet length> <count of re-tx>
<drop reason> <freq. channel>
      # event: Event type.
      event_type=$2
      # packet_type: Packet type.
      packet_type=$5
      # source: Node ID that starts the transmission.
      source=substr($8,2,length($8))
      # destination: Node ID to whom the packets are addressed.
      destination=substr($10,1,length($10)-1)
      # start_time: Time (in tics) when the transmission is started.
      start_time=$3
      # packet_id: Packet's unique id.
      packet_id=$11
      # packet_length: Length of the packet (counting the headers).
      packet_length=$12

    if( start_time > (tics_per_second*interval*interval_size) ) {
      printf ("%d-%d;%d;%d;%d;%d;%d;%d;%d;%d;\n", (interval-
1)*interval_size, interval*interval_size, collisions_f1, collisions_f2,
tx_f1, tx_f2, rtx_f1, rtx_f2,raros_f1,raros_f2)
        collisions_f1=0
        collisions_f2=0
      interval++
```

```
        tx_f1=0
        tx_f2=0
        rtx_f1=0
        rtx_f2=0
        raros_f1=0
        raros_f2=0
    }

    if (packet_type == "DATA" && event_type == "TX" && packet_length ==
"570")
    {
      if(source == "5")
      {
            if ( !( packet_id in F1 ) )
                  {
                        F1[packet_id] = 1
            }
      }

      if(source == "6")
      {
            if ( !( packet_id in F2 ) )
                  {
                        F2[packet_id] = 1
            }
      }
    }

    if ( packet_type == "DATA" && packet_length == "570" && event_type ==
"TX")
    {
      if ( packet_id in F1 ) {
          tx_f1++
      } else if ( packet_id in F2 ) {
            tx_f2++
      }
    }

    if ( packet_type == "DATA" && packet_length == "570" && event_type ==
"RTX")
    {
      if ( packet_id in F1 ) {
          rtx_f1++
      } else if ( packet_id in F2 ) {
            rtx_f2++
      }
    }

    if ( packet_type == "DATA" && packet_length == "570" && $14 == "COLL")
    {
      if ( packet_id in F1 ) {
          collisions_f1++
      } else if ( packet_id in F2 ) {
            collisions_f2++
      }
    }
}

END{
```

```
    printf ("%d-%d;%d;%d;%d;%d;%d;%d;%d;%d;\n", (interval-1)*interval_size,
interval*interval_size, collisions_f1, collisions_f2, tx_f1, tx_f2, rtx_f1,
rtx_f2)
    close(output)
}
```

The filter is invoked with the following command.

```
awk -f collision_filter.awk < results.tr
```

Here is an example of the output of the filter.

```
[nctuns@fedora COLISIONES]$ awk -f collision_filter.awk < results.tr
INTERVAL;COLL_F1;COLL_F2;TX_F1;TX_F2;RTX_F1;RTX_F2;
20-25;2933;904;18887;11803;1791;1022;15942;
25-30;2;23;3596;3286;59;191;3365;
30-35;0;2;97;1724;673;426;3819;
35-40;0;0;60;38;427;267;3621;
40-45;0;0;0;0;0;0;4380;
45-50;0;0;0;0;0;0;3963;
50-55;0;0;0;0;0;0;3033;
55-60;0;0;0;0;0;0;2628;
60-65;0;0;0;0;0;0;3213;
65-70;0;0;0;0;0;0;3163;
70-75;0;0;0;0;0;0;3207;
75-80;0;0;0;0;0;0;2643;
```

9. References and bibliography

[1] **Real Automóvil Club de España,** http://www.race.es/seguridad_vial/estudios_informes/datos_siniestralidad/

[2] **Olariu, S. and Weigle, M. C.,** *Vehicular Networks, From Theory To Practice* ,CRC Press, Taylor & Francis Group, 2009.

[3] **Guo, H.,** *Automotive Informatics Communicative Systems: Principles in Vehicular Networks and Data Exchange*, Premier Reference Source, 2009.

[4] **eSafetyAware,** "www.esafetyaware.eu"

[5] **Elson, J. and Estrin, D.,** *Sensor Networks: A Bridge to the Physical World*, Wireless Sensor Networks, Boston, Kluwe Academic Publisher, 2004.

[6] **Sohraby, K., Minoli, D. and Znati, T.,** *Wireless Sensor Networks. Technology, Protocols and Applications*, John Wiley & Sons, Inc., 2007.

[7] **IEEE, Computer Society,** *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPAN's)*, New York, 2006.

[8] **Ortuño Perez, M.A.,** *Protocolo de encaminamiento en origen con identificadores no únicos para redes Ad-Hoc de dispositivos con recursos limitados,* Tesis Doctoral, Universidad Rey Juan Carlos, Madrid, 2006.

[9] **Garcia Davis, E.J.,** *Implementación de Protocolos de Transporte en Redes de Sensores,* Proyecto Final de Máster, UPC, Barcelona, 2009.

[10] **Crossbow Technology**, http://www.xbow.com/

[11] **Crossbow Technology products:** http://www.xbow.com/defense-solutions/products/index.html

[12] **Weingärtner, E. and Kargl, F.***, A Prototype Study on Hybrid Sensor Vehicular Networks,* RWTH Aachen, Aachen, Germany, 2007.

[13] **Jiang, D. and Delgrossi, L.***, IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environment,* 67th IEEE Vehicular Technology Conference VTC, Singapore, 2008.

[14] **Aguilar, M.,** *Redes Ad-hoc Multisalto y Redes de Vehículos,* Transparencias de clase, Barcelona UPC, 2009.

[15] **Bernsen, J. and Manivannan, D.,** *Unicast routing protocols for vehicular ad-hoc networks: A critical comparison and classification,* Pervasive and Mobile Computing, Elsevier B.V., Vol. 5-1, pp. 1-18, 2009.

[16] **Buruhanudeen, S., Othman, M., Mohd Ali, B., & Othman, M.,** *Performance Comparison of MANET Associativity Based Routing (ABR) and the Improvisation Done for a More Realiable and Efficient Routing,* The 3rd Information and Communication Technology Seminar (ICTS), Surabaya, Indonesia, 2007.

[17] **Buruhanudeen, S., Othman, M., Othman, M., & Mohd Ali, B.,** *Existing MANET Routing Protocols and Metrics used Towards the Efficiency and Reliability -An Overview,* Telecommunications and Malaysia International Conference on Communications, ICT-MICC, 2007.

[18] **Camp, T., Boleng, J., Williams, B., Wilcox, L., & Navidi, W.,** *Performance Evaluation of Two Location Based Routing Protocols*, IEEE 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), pp. 1678-1687, 2002.

[19] **Chakeres, I. and Perkins, *C.,*** *Dynamic MANET On-demand (DYMO) Routing, 2009.* http://tools.ietf.org/html/draft-ietf-manet-dymo-17.

[20] **Corson, M. S., Papademetriou, S., Papadopoulos, P., Park, V., & Qayyum, A.,** *An Internet MANET Encapsulation Protocol (IMEP) Specification, 1999.* http://tools.ietf.org/html/draft-ietf-manet-imep-spec-01.

[21] **Djenouri, D., Derhab, A. and Badache, N.,** *Ad-hoc Networks Routing Protocols and Mobility,* The International Arab Journal of Information Technology, Vol. 3, 4, 2006.

[22] ***Djenouri, D., Soualhi, W. and Nekka, E.,*** *Mobility Models in Vehicular Ad-hoc Networks: The Overtaking Impact,* The Mediterranean Journal of Computers and Networks, 2008.

[23] **European Commission Transport**. *2009,* http://ec.europa.eu/transport/galileo/index_en.htm

[24] **Haas, Z., Pearlman, M. and Samar, P.,** *The Zone Routing Protocol (ZRP) for Ad-hoc Networks, 2002.* http://tools.ietf.org/id/draft-ietf-manet-zone-zrp-04.txt.

[25] **Hofman-Wellwnhof, B., Lichtenegger, H. and Collins, J.,** *Global Positioning System: Theory and Practice,* Springer, 5th Ed., 2001.

[26] **Jhonson, D.B. and Maltz, D.A.,** *Dynamic Source Routing in Ad-hoc Wireless, Mobile Computing*, Imielinski and H. Korth, Eds. Kluwer Academic Publisher, pp. 153-181, 1996.

[27] **Kaplan, E. and Hegarty, C.,** *Understanding GPS: Principles and Applications,* Boston, Artech Hourse, 2nd Ed., 2006.

[28] **Kleinrock, L. and Stevens, K.,** *Fisheye: a lenslike computer display transformation,* UCLA, Computer Science Department, Technical Report, 1971.

[29] **Ko, Y.B. and Vaidya, N.,** *Location-Aided Routing (LAR) in Mobile Ad-hoc Networks,* International Conference on Mobile Computing and Networking (Mobicom 98), 1998.

[30] **Larsson, T. and Hedman, N.,** *Routing Protocols in Wireless Ad-hoc Networks - A Simulation Study,* Master's Thesis, Lulea University of Technology, Stockholm, 1998.

[31] **Leontiadis, I. and Mascolo, C.** *GeOpps: Geographical Opportunistic Routing for Vehicular Networks,* World of Wireless, Mobile and Multimedia Networks, WoWMoM 2007, 2007.

[32] **Li, F. and Wang, Y.** *Routing in Vehicular Ad-hoc Networks: A survey,* IEEE Vehicular Technology Magazine, 2007.

[33] **Lochert, C., Hartenstein, H., Tian, J., Füßler, H., Hermann, D., & Mauve, M.,** *A Routing Strategy for Vehicular Ad-hoc Networks in City,* IEEE, Intelligent Vehicles Symposium '03, pp. 156-161, 2003.

[34] **Lochert, C., Mauve, M., Füßler, H., & Hartenstein, H.,** *Geographic Routing in City Scenarios,* ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 9, pp. 69-72, 2005.

[35] **Mo, Z., Zhu, H., Makki, K., & Pissinou, N.,** *Muru: A multi-hop routing protocol for urban vehicular ad-hoc networks",* 3rd Annual International Conference on Mobile and Ubiquitous Systems – Workshops, San Jose, CA, 2006.

[36] **Murty, S. and Garcia-Luna-Aceves, J.J.,** *An Efficient Routing Protocol for Wireless Networks,* Mobile Networks and Aplication, pp. 183-197, 1996.

[37] **Namboodiri, V. and Gao, L.,** *Prediction-based routing for vehicular ad-hoc networks,* IEEE transactions on vehicular technology, Vol. 56, pp. 2332-2345, 2007.

[38] **Naumov, V. and Gross, T.,** *Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks,* IEEE INFOCOM'07, 2007.

[39] **Niculescu, D. and Nath, B.,** *Trajectory Based Forwarding and Its Applications,* 9th International Conference on Mobile Computing and Networking (MOBICOM '03), pp. 260-272, 2003.

[40] **Park, V. and Corson, M.,** *A Performance Comparison of the Temporally-Ordered Routing Algorithm and Ideal Link-State Routing,* IEEE International Symposium of Systems and Communications, 1998.

[41] **Pei, G., Gerla, M. and Chen, T.,** *Fisheye State Routing in Mobile Ad-hoc Network,* IEEE International Conference ons Communications (ICC), New Orleans, pp. 70-74, 2000.

[42] **Perkins, C.,** *Ad-hoc On-Demand Distance Vector (AODV) Routing,* 2003. http://www.ietf.org/rfc/rfc3561.txt.

[43] **Perkins, C. and Bhagwat, P.,** *Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,* SIGCOM '94 Conference on Communications Architecture, Protocols and Application, England, pp. 234-24, 1994.

[44] **See, S., Liu, G., Lee, B., Foh, C., Wong, K., and Lee, K.,** *A-STAR: A Mobile Ad-hoc Routing Strategy for Metropolis,* 3rd International Networking Conference IFIP-TC6 (IFIP '04), 2004.

[45] **The Internet Engineering Task Force**. http://www.ietf.org/. 1992

[46] **Tian, J., Han, L. and Rothermel, K.,** *Spatially Aware Packet Routing for Mobile Ad-hoc Inter-Vehicle Radio Networks*, IEEE Intelligent Transportation Systems, Shanghai, China, pp. 1546-1551, 2003.

[47] **Yi, Y., Lee, S., Su, W., and Gerla, M.,** *On-Demand Multicast Routing Protocol (ODMRP) for Ad-hoc Networks,* 2002. http://tools.ietf.org/html/draft-ietf-manet-odmrp-04.

[48] **Zhao, J. and Cao, J.,** *VADD: Vehicle-assisted data delivery in vehicular ad-hoc networks,* 25th IEEE International Conference on Computer Communications, INFOCOM 2006, pp.1-12, 2006.

[49] **Al-Karaki, J. N. and Kamal, A. E.,** *Routing Techniques in Wireless Sensor Networks: A survey,* IEEE Wireless Communications, 2004.

[50] **Hassan, A.,** *VANET simulation*, Master's Thesis in Electrical Engineering, Halmstad University, Sweden, 2009.

[51] **Weingartner, E., vom Lehn, H., Wehrle, K.,** *A performance comparison of recent network simulators*, ICC '09 IEEE International Conference on Communications, 2009.

[52] **Rajabi, A., Aguilar, M., Tripp, C.**, *Performance Evaluation of Vehicular Ad-hoc Networks using simulation tools*, Proyecto de Fin de Carrera, UPC, 2010.

[53] **Diseño e implementación de un servidor de vídeo adaptativo en simulador de redes ad-hoc**, Muñoz, C., Aguilar, M., Proyecto de fin de carrera, UPC, 2009

[54] "The Protocol Developer Manual for the NCTUns 6.0 Network Simulator and Emulator", http://nsl10.csie.nctu.edu.tw/support/documentation/DeveloperManual.pdf , S.Y. Wang, C.L. Chou, C. C. Lin, and C.H. Huang, 2010.

[55] The GNU Awk User's Guide http://www.gnu.org/manual/gawk/gawk.html

[56] **Moustafa, H. and Zhang, Y.,** Vehicular *Networks: Techniques, Standards, and Applications*, CRC Press, 2009.