



Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROYECTO FINAL DE MÁSTER

Performance Evaluation of Bandwidth-Aware Routing Protocol in Urban Scenarios for Vehicular Ad hoc NETWORKS

Estudios: Máster en Ingeniería Telemática

Autor: Nely Patricia López Márquez

Directora: Mónica Aguilar Igartua

Co-directora: Carolina Tripp Barba

Barcelona, Abril 2012

*"Nunca desistas de un sueño.
Solo trata de ver las señales que te lleven a él"*

Paulo Coelho.

AGRADECIMIENTOS

Primeramente a Dios, por darme la dicha de ver culminada esta etapa de mi vida con la que tanto soñaba. Por ayudarme a crecer no sólo profesionalmente sino también personal y espiritualmente. Por todo lo que ésta experiencia de estar lejos significa y me ha enseñado. Por darme la fuerza de mantenerme en pie durante todo este tiempo y seguir mirando siempre hacia adelante a pesar de las dificultades y los obstáculos.

A mi familia; a mi mamá, a mi papá, a mi hermana. Que han soportado conmigo el tiempo y la distancia. A mi hermana por enseñarme con su ejemplo que con fuerza y con valentía se logran los buenos propósitos. A mi mamá por escucharme siempre, por estar pendiente de mí todos los días. Por apoyarme en este sueño y querer siempre mi bienestar a pesar de estar lejos de tí. Por rezar por mí, por soportar mi mal genio en momentos de desesperación. Sé que a veces no encontrabas palabras para animarme, pero siempre estabas allí, escuchándome. A mi papá por apoyarme en este sueño. Por apoyarme siempre en lo que he decidido hacer, por no cortarme las alas y dejarme volar a donde he querido ir. Por tus consejos y por siempre buscar lo mejor para mí.

A mi prometido Mario, que también eres mi familia. Por tu apoyo incondicional siempre en las buenas, en las malas y en las peores. Por ser mi apoyo fundamental en cumplir este sueño que también es tuyo. Por estar pendiente de mí día a día durante todo este tiempo sin que faltara ninguno. Por escucharme siempre, por tus palabras de aliento siempre que sentía que ya no podía más con la presión, por soportar mi mal humor cuando me sentía desesperada. Por confiar en mi capacidad cuando ni yo misma confiaba en ella. Por siempre ayudarme a encontrar soluciones a todo lo que se me presentaba. Gracias por hacerme feliz.

A Mónica Aguilar, por su constante, motivación, entusiasmo y supervisión en el desarrollo de este proyecto. Gracias por responder al instante mis mails.

A Carolina Tripp, por tu constante apoyo y atención, que siempre que tenía alguna duda y la consultaba contigo, siempre me dabas tiempo y me orientabas a pesar de estar ocupada. Gracias jefa.

Nely.

INDEX

INDEX OF FIGURES.....	8
INDEX OF TABLES.....	11
GLOSSARY OF ACRONYMS.....	12
ABSTRACT.....	15
RESUMEN.....	16
RESUM.....	17
1. INTRODUCTION AND OBJECTIVES.....	18
1.1 Introduction.....	18
1.2 Objectives.....	20
2. INTRODUCTION TO MOBILE AND VEHICULAR AD HOC NETWORKS.....	21
2.1 Overview to Mobile Ad hoc Networks (MANETs).....	22
2.1.1 Mobile Wireless Networks.....	22
2.1.1.1 Infrastructure Wireless Networks.....	23
2.1.1.2 Infrastructure-less Wireless Networks.....	23
2.1.2 Ad hoc Networks Characteristics.....	24
2.1.3 Ad hoc Network Types.....	25
2.1.3.1 Wireless Sensor Networks (WSNs).....	25
2.1.3.2 Mobile Ad hoc NETWORKS (MANETs).....	26
2.2 State of the Art in Vehicular Ad hoc Networks (VANETs).....	27
2.2.1 VANETs Architecture.....	28
2.2.2 VANETs Characteristics.....	28
2.2.3 VANETs Applications and Services.....	29
2.2.4 Challenges in VANETs	30
2.2.4.1 Technical Challenges.....	31
2.2.4.1.1 Reliable Communications and MAC Protocols.....	31
2.2.4.1.2 Routing and Dissemination.....	32
2.2.4.1.3 Security.....	32
2.2.4.1.4 IP Configuration and Mobility Management.....	32
2.2.4.1.5 Business Models.....	33
2.3 Standardization and Research Projects Initiatives.....	33
2.3.1 Dedicated Short-Range Communications (DSRC) Technology.....	33

2.3.2 CAR-2-CAR Communication Consortium.....	33
2.3.3 WAVE (Wireless Access in Vehicular Environments) stack.....	34
2.3.4 CARLINK.....	35
2.3.5 PReVENT.....	35
2.3.6 Network on Wheels (NOW).....	35
2.3.7 SAFESPOT.....	36
2.3.8 Cooperative Vehicle-Infrastructure Systems (CVIS).....	36
2.3.9 Privacy Enabled Capability In Co-Operative Systems and Safety Applications (PRECIOSA).....	37
2.3.10 Secure Vehicle Communication (SEVECOM).....	37
3. ROUTING PROTOCOLS FOR VEHICULAR AD HOC NETWORKS (VANETs).....	39
3.1 Background Routing for VANETs	39
3.1.1 Taxonomy of Routing Protocols.....	40
3.1.1.1 Topology-based Routing.....	40
3.1.1.2 Position-based Routing.....	41
3.1.1.3 Movement-based Routing.....	42
3.1.1.4 Map-based Routing.....	42
3.1.1.5 Traffic-aware Routing.....	43
3.2 Quality of Service Routing for VANETs	45
3.2.1 Available Bandwidth (AB).....	46
3.3 Ad hoc On-demand Distance Vector (AODV).....	48
3.3.1 AODV Operation.....	49
3.3.1.1 AODV Route Discovery.....	49
3.3.1.2 AODV Route Maintenance.....	50
3.4 Available Bandwidth Estimation (ABE).....	50
3.4.1 ABE Operation.....	51
3.4.1.1 Estimating Nodes Emission Capabilities.....	51
3.4.1.2 Estimating Link Available Bandwidth.....	52
3.4.1.3 Estimating the Collisions Probability	53
3.4.1.4 Estimating the Exponential Backoff Time.....	54
3.4.2 ABE Calculation.....	54
4. SIMULATION ENVIRONMENT NCTUns AND CITYMOB.....	56
4.1 Introduction to NCTUns.....	56
4.1.1 Capabilities and Features.....	56

4.1.2 Architecture and Components.....	60
4.2 Installation and Execution of NCTUns.....	62
4.2.1 Installation Steps.....	62
4.2.2 Starting NCTUns.....	65
4.2.2.1 Operation Modes.....	66
4.3 Design VANET urban scenario.....	66
4.4 Introduction to Citymob.....	68
4.4.1 Mobility Models.....	68
4.4.2 Citymob Guidelines.....	69
4.4.3 Installation and Operation.....	69
4.5 Design VANET urban movement.....	70
4.6 Configuration of Simulation Environment Scenarios.....	72
5. SIMULATIONS AND RESULTS.....	84
5.1 Simulation Settings.....	84
5.2 Scenario, Simulations and Graphs Description.....	85
5.3 Metrics Analyzed.....	86
5.4 Scenario with 20 vehicles.....	87
5.4.1 Packet Loss Ratio.....	88
5.4.2 Average Delay per Packet.....	89
5.4.3 Throughput.....	90
5.4.4 Throughput Over Time.....	92
5.4.5 Packet Delivery Ratio.....	94
5.5 Scenario with 40 vehicles.....	95
5.5.1 Packet Loss Ratio.....	95
5.5.2 Average Delay per Packet.....	96
5.5.3 Throughput	97
5.5.4 Throughput Over Time.....	99
5.5.5 Packets Delivery Ratio.....	100
5.6 Scenario with 60 vehicles.....	101
5.6.1 Packets Loss Ratio.....	102
5.6.2 Average Delay per Packet.....	103
5.6.3 Throughput.....	104
5.6.4 Throughput Over Time.....	105
5.6.5 Packets Delivery Ratio.....	106

5.7 Scenario with 80 vehicles.....	107
5.7.1 Packets Loss Ratio.....	108
5.7.2 Average Delay per Packet.....	109
5.7.3 Throughput.....	110
5.7.4 Throughput Over Time.....	112
5.7.5 Packet Delivery Ratio.....	114
5.8 Scenario with 100 vehicles.....	114
5.8.1 Packets Loss Ratio.....	115
5.8.2 Average Delay per Packet.....	116
5.8.3 Throughput.....	117
5.8.4 Throughput Over Time.....	119
5.8.5 Packets Delivery Ratio.....	120
6. CONCLUSIONS AND FUTURE WORK.....	122
6.1 Conclusions.....	122
6.2 Future Work.....	124
ANNEX 1. AWK FILTER.....	125
REFERENCES.....	138

INDEX OF FIGURES

<i>Figure 2.1. Cellular network / WLAN.....</i>	<i>23</i>
<i>Figure 2.2. Ad hoc network.....</i>	<i>24</i>
<i>Figure 2.3. VANET environment in urban scenario.....</i>	<i>27</i>
<i>Figure 2.4. VANETs architecture.....</i>	<i>28</i>
<i>Figure 2.5. IEEE WAVE stack.....</i>	<i>34</i>
<i>Figure 3.1. Taxonomy of routing protocols in VANETs.....</i>	<i>40</i>
<i>Figure 3.2. Junctions and paths to forward packets in GSR, SAR and JARR.....</i>	<i>43</i>
<i>Figure 3.3. RREQ packets flooding AODV.....</i>	<i>49</i>
<i>Figure 3.4. RREP packets in AODV.....</i>	<i>50</i>
<i>Figure 4.1. NCTUns generic architecture.....</i>	<i>60</i>
<i>Figure 4.2. NCTUns distributed architecture.....</i>	<i>62</i>
<i>Figure 4.3. GRUB bootloader.....</i>	<i>63</i>
<i>Figure 4.4. Login screen nctuns user on fedora.....</i>	<i>64</i>
<i>Figure 4.5. Screenshot NCTUns.....</i>	<i>65</i>
<i>Figure 4.6. Operation modes.....</i>	<i>66</i>
<i>Figure 4.7. ITS crossroad.....</i>	<i>67</i>
<i>Figure 4.8. ITS road segment.....</i>	<i>67</i>
<i>Figure 4.9. Obstacles.....</i>	<i>67</i>
<i>Figure 4.10. Screenshot scenario: streets, intersections and buildings.....</i>	<i>67</i>
<i>Figure 4.11. Scenario dimensions.....</i>	<i>68</i>
<i>Figure 4.12. Import vehicles and movement.....</i>	<i>72</i>
<i>Figure 4.13. Screenshot scenario with 100 vehicles.....</i>	<i>73</i>
<i>Figure 4.14. Edit property mode.....</i>	<i>73</i>
<i>Figure 4.15. Protocol stack in the Node Editor.....</i>	<i>74</i>
<i>Figure 4.16. C.P.A.N.S.T. button.....</i>	<i>75</i>
<i>Figure 4.17. Command stg application program.....</i>	<i>76</i>
<i>Figure 4.18. Add, modify and delete command application program.....</i>	<i>77</i>
<i>Figure 4.19. Command rtg application program.....</i>	<i>77</i>
<i>Figure 4.20. Add, modify and delete command application program.....</i>	<i>78</i>
<i>Figure 4.21. Physical layer button.....</i>	<i>78</i>
<i>Figure 4.22. Specify physical layer and channel model parameter.....</i>	<i>79</i>

Figure 4.23. Nodes to be selected.....	79
Figure 4.24. Simulation time and random number seed.....	80
Figure 4.25. Dispatcher configuration.....	80
Figure 4.26. Run simulation mode.....	81
Figure 4.27. Message simulation is done.....	81
Figure 4.28. Packet animation player.....	81
Figure 5.1. Screenshot scenario with 20 vehicles.....	88
Figure 5.2. End-to-end average packet loss ratio with 20 vehicles.....	89
Figure 5.3. End-to-end average delay per packet with 20 vehicles.....	90
Figure 5.4. End-to-end average throughput with 20 vehicles.....	92
Figure 5.5. Throughput over time with 20 vehicles for AODV.....	93
Figure 5.6. Throughput over time with 20 vehicles for ABE.....	94
Figure 5.7. End-to-end average packet delivery ratio with 20 vehicles.....	94
Figure 5.8. Screenshot scenario with 40 vehicles.....	95
Figure 5.9. End-to-end average packet loss ratio with 40 vehicles.....	96
Figure 5.10. End-to-end average delay per packet with 40 vehicles.....	97
Figure 5.11. End-to-end average throughput with 40 vehicles.....	98
Figure 5.12. Throughput over time with 40 vehicles for AODV.....	99
Figure 5.13. Throughput over time with 40 vehicles for ABE.....	100
Figure 5.14. End-to-end average packet delivery ratio with 40 vehicles.....	100
Figure 5.15. Screenshot scenario with 60 vehicles.....	101
Figure 5.16. End-to-end average packet loss ratio with 60 vehicles.....	102
Figure 5.17. End-to-end average delay per packet with 60 vehicles.....	103
Figure 5.18. End-to-end average throughput with 60 vehicles.....	104
Figure 5.19. Throughput over time with 60 vehicles for AODV.....	105
Figure 5.20. Throughput over time with 60 vehicles for ABE.....	106
Figure 5.21. End-to-end average packet delivery ratio with 60 vehicles.....	107
Figure 5.22. Screenshot scenario with 80 vehicles.....	108
Figure 5.23. End-to-end average packet loss ratio with 80 vehicles.....	109
Figure 5.24. End-to-end average delay per packet with 80 vehicles.....	110
Figure 5.25. End-to-end average throughput with 80 vehicles.....	111
Figure 5.26. Throughput over time with 80 vehicles for AODV.....	112
Figure 5.27. Throughput over time with 80 vehicles for ABE.....	113
Figure 5.28. End-to-end average packet delivery ratio with 80 vehicles.....	114

<i>Figure 5.29. Screenshot scenario with 100 vehicles.</i>	<i>115</i>
<i>Figure 5.30. End-to-end average packet loss ratio with 100 vehicles.</i>	<i>116</i>
<i>Figure 5.31. End-to-end average delay per packet with 100 vehicles.</i>	<i>117</i>
<i>Figure 5.32. End-to-end average throughput with 100 vehicles.</i>	<i>118</i>
<i>Figure 5.33. Throughput over time with 100 vehicles for AODV.</i>	<i>119</i>
<i>Figure 5.34. Throughput over time with 100 vehicles for ABE.</i>	<i>120</i>
<i>Figure 5.35. End-to-end packet delivery ratio with 100 vehicles.</i>	<i>121</i>

INDEX OF TABLES

<i>Table 2.1. Examples of applications and requirements.....</i>	<i>30</i>
<i>Table 5.1. Simulation Settings.....</i>	<i>85</i>
<i>Table 5.2. Number of packets Tx, Rx and lost with 20 vehicles.....</i>	<i>91</i>
<i>Table 5.3. Number of packets Tx, Rx and lost with 40 vehicles.....</i>	<i>97</i>
<i>Table 5.4. Number of packets Tx, Rx and lost with 60 vehicles.....</i>	<i>104</i>
<i>Table 5.5. Number of packets Tx, Rx and lost with 80 vehicles.....</i>	<i>111</i>
<i>Table 5.6. Number of packets Tx, Rx and lost with 100 vehicles.....</i>	<i>118</i>

GLOSSARY OF ACRONYMS

A-STAR	<i>Anchor Based Street and Traffic Aware Routing</i>
AB	<i>Available Bandwidth</i>
ABE	<i>Available Bandwidth Estimation</i>
ACC	<i>Adaptive Admission Control</i>
AODV	<i>Ad hoc On-demand Distance Vector</i>
AP	<i>Access Point</i>
ARPANET	<i>Advanced Research Projects Agency NETwork</i>
BRuIT	<i>Bandwidth Reservation under InTerference influence</i>
BS	<i>Base Station</i>
BWER	<i>Bandwidth Efficiency Ratio</i>
C.P.A.N.S.T	<i>Copy the nodes Protocol stack to All Nodes of the Same Type</i>
CACP	<i>Contention-aware Admission Control Protocol</i>
CAR	<i>Connectivity Aware Routing</i>
CAR-2-CAR	<i>CAR-2-CAR Communication Consortium</i>
CARC	<i>Call Admission and Rate Control</i>
CBR	<i>Constant Bit Rate</i>
CI	<i>Confidence Interval</i>
CVIS	<i>Cooperative Vehicle-Infrastructure Systems</i>
CW	<i>Contention Window</i>
D.I.R.	<i>Data Interference Range</i>
D.T.R.	<i>Data Transmission Range</i>
DARPA	<i>Defense Advanced Research Project Agency</i>
DGR	<i>Directional Greedy Forwarding</i>
DIFS	<i>Distributed Inter-Frame Space</i>
DM	<i>Downtown Model</i>
DoD	<i>Department of Defense</i>
DSDV	<i>Destination-Sequenced Distance-Vector Routing</i>
DSR	<i>Dynamic Source Routing</i>
DSRC	<i>Dedicated Short Range Communications</i>
EDD	<i>Expected Disconnection Degree</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FAT	<i>Fraction of Air Time</i>
GNU/GPL	<i>GNU/General Public License</i>

GPCR	<i>Greedy Perimeter Coordinator Routing</i>
GPRS	<i>General Packet Radio Services</i>
GPS	<i>Global Positioning System</i>
GPSR	<i>Greedy Perimeter Stateless Routing</i>
GRUB	<i>GRand Unified Bootloader</i>
GSM	<i>Global System for Mobile communications</i>
GSR	<i>Geographic Source Routing</i>
GUI	<i>Graphic User Interface</i>
IAB	<i>Improved Available Bandwidth</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
IP	<i>Internet Protocol</i>
ITS	<i>Intelligent Transportation Systems</i>
ITU	<i>International Telecommunication Union - T</i>
JARR	<i>Junction Based Adaptive Reactivate Routing</i>
LAN	<i>Local Area Network</i>
MAC	<i>Medium Access Control</i>
MAGF	<i>Movement Aware Greedy Forwarding</i>
MANET	<i>Mobile Ad hoc Network</i>
MIB	<i>Management Information Base</i>
MM	<i>Manhattan Model</i>
NCTUns	<i>National Chiao Tung University network simulator</i>
NLOS	<i>Non Light Of Sight</i>
NOW	<i>Network On Wheels</i>
OBU	<i>On Board Unit</i>
OLSR	<i>Optimized Link State Routing</i>
PDGR	<i>Predictive Directional Greedy Routing</i>
PGA	<i>Probe Gap Models</i>
PRECIOSA	<i>Privacy Enabled Capability In Co-Operative Systems and Safety Applications</i>
PRM	<i>Probe Rate Models</i>
PRNet	<i>Packet Radio Network</i>
QoS	<i>Quality of Service</i>
RABE	<i>Retransmission-based Available Bandwidth Estimation</i>
RBVT	<i>Road-Based with Vehicular Traffic protocols</i>
RERR	<i>Route Error</i>

RREP	<i>Route Reply</i>
RREQ	<i>Route Request</i>
RSU	<i>Road Side Unit</i>
Rx	<i>Received</i>
SAR	<i>Spatially Aware Routing</i>
SELinux	<i>Security Enhanced Linux</i>
SEVECOM	<i>Secure Vehicle Communications</i>
SM	<i>Simple Model</i>
TC	<i>Technical Committee</i>
TCP	<i>Transmission Control Protocol</i>
TTL	<i>Time To Life</i>
Tx	<i>Transmitted</i>
UAV	<i>Unmanned Airborne Vehicles</i>
UDP	<i>User Datagram Protocol</i>
UMTS	<i>Universal Mobile Telecommunications Systems</i>
V2R	<i>Vehicle-to-RSU</i>
V2V	<i>Vehicle-to-Vehicle</i>
VANET	<i>Vehicular Ad hoc Network</i>
WAVE	<i>Wireless Access in Vehicular Environments</i>
WG	<i>Working Group</i>
WLAN	<i>Wireless Local Area Network</i>
WSN	<i>Wireless Sensor Network</i>

ABSTRACT

Nowadays, there are many existing technologies designed to provide vehicular road travel safer and comfortable, by using geographical positioning systems and multimedia applications. The current requirements of transmission information of these technologies, requires great demand on algorithms which often perform less than optimally, especially when they have to interact with other vehicles to disseminate information.

Some services are critical and require to guaranteeing a certain level of Quality of Service (QoS) on metrics such as low delay, packet loss, bandwidth among others. This issue is a big challenge in VANETs environments where the high vehicles mobility is a parameter non-measurable especially in urban scenarios, where vehicles follow a movement pattern following streets at random directions. As a consequence, routing protocols have the hard task of determining and establishing the forwarding path to send data from source to destination via multihop steps using intermediate vehicles in an efficient way.

The main goal of this project is to carry out a performance evaluation of a new routing protocol developed in [Sarr 2008] which is able to guarantee certain level of QoS (Quality of service) called ABE (Available Bandwidth Estimation) in comparison to a well-known routing protocol such as AODV (Ad hoc On-demand Distance Vector). A novelty in this project is the analysis of this approach in vehicular ad hoc networks, to see what are the benefits of applying a QoS-aware routing protocol in such networks.

RESUMEN

Hoy en día, existen muchas tecnologías diseñadas para proporcionar desplazamientos por carretera más seguros y confortables, mediante el uso de sistemas de posicionamiento geográfico y aplicaciones multimedia. Los actuales requerimientos de transmisión de información de éstas tecnologías, requieren gran demanda de algoritmos los cuales a menudo rinden menos que el desempeño óptimo, especialmente cuando tienen que interactuar con otros vehículos con el fin de difundir información.

Algunos servicios son críticos y requieren garantizar un cierto nivel de Calidad de Servicio (QoS) en métricas tales como retardo, pérdida de paquetes, ancho de banda, entre otras. Esta cuestión es un gran desafío dentro de los entornos VANETs donde la alta movilidad de los vehículos es un parámetro no cuantificable especialmente en escenarios urbanos, donde los vehículos siguen un patrón de movimiento siguiendo las calles en direcciones aleatorias. Como consecuencia, los protocolos de enrutamiento tienen la difícil tarea de determinar y establecer la ruta de transmisión para enviar los datos desde el origen hacia el destino vía multisaltos utilizando vehículos intermedios de una manera eficiente.

El principal objetivo de este proyecto es realizar una evaluación del rendimiento de un nuevo protocolo de enrutamiento desarrollado en [Sarr 2008] el cual es capaz de garantizar un cierto nivel de Calidad de Servicio (QoS) llamado ABE (Available Bandwidth Estimation) en comparación con un ampliamente conocido protocolo de enrutamiento AODV (Ad hoc On-demand Distance Vector). La novedad en este proyecto es el análisis de este esquema en redes vehiculares ad hoc, para ver cuáles son los beneficios de aplicar un protocolo de enrutamiento QoS-aware en este tipo de redes.

RESUM

A dia d'avui hi ha moltes tecnologies dissenyades per proporcionar als vehicles desplaçaments més còmodes i més segurs mitjançant l'ús de geoposicionament per satèl·lit i aplicacions multimèdia. Els actuals requeriments de transmissió d'informació d'aquestes tecnologies suposa un alt ús d'algoritmes que sovint són poc òptims, especialment quan aquests han d'interactuar amb altres vehicles per disseminació de la informació.

Alguns serveis són crítics i és necessari garantir un cert nivell de qualitat del servei (QoS) en mètriques tals com el retard, la pèrdua de paquets i l'ample de banda, entre d'altres. Aquesta qüestió suposa un gran desafiament en l'entorn de les VANETs, on l'alta mobilitat dels vehicles és un paràmetre no quantificable especialment en escenaris urbans, on els vehicles segueixen un patró de moviment aleatori pels carrers de la ciutat. Com a conseqüència, els protocols d'enrutament tenen la difícil tasca de determinar i establir la ruta de transmissió per enviar les dades des de l'origen fins al destí via multi-salt utilitzant vehicles intermitjos d'una manera eficient.

El principal objectiu d'aquest projecte és realitzar una avaluació del rendiment d'un nou protocol d'enrutament desenvolupat a [Sarr 2008] el qual és capaç de garantir una certa qualitat del servei (QoS), anomenat ABE (Available Bandwidth Estimation), en comparació amb el protocol d'enrutat AODV (Ad hoc On-demand Distance Vector). La novetat d'aquest projecte és l'anàlisi d'aquest esquema en xarxes vehiculars ad hoc, per veure quins són els beneficis d'aplicar un protocol d'enrutat QoS-aware en xarxes d'aquest tipus.

1. INTRODUCTION AND OBJECTIVES

1.1 Introduction

In recent years, advances in wireless networks have allowed and carried out the development and introduction of a new kind of wireless networks called Vehicular Ad hoc Networks. The term *Vehicular Ad hoc NETWORKS (VANETs)* arises from a special form of *Mobile Ad hoc NETWORKS (MANETs)*. VANETs have the main objective to provide us new systems to enhance drivers and passenger's safety and comfort.

VANETs have nature of being distributed and self-organizing networks formed among moving vehicles which are equipped with wireless communication devices. This kind of networks is developed as part of *Intelligent Transportation Systems* [ITS Spain] [ITS America]. One of the main goals of ITS is to improve transport outcomes such as increase transport safety, reduce traffic congestion, increase travel reliability, reduce waiting times, reduce air pollution and fuel consumption, among other objectives.

The integration of navigation systems, digital maps, and wireless communication devices together helps to develop numerous kinds of applications for ITS to improve safety roads. The up to date

information provided by all these systems helps drivers to obtain real time information about road conditions. For instance, a simple warning messages sent by vehicles involved in an accident enhances traffic safety and congestion by helping drivers to take an alternate road before entering on crash zone and avoiding to take routes in case of congestion, thus saving time and fuel consumption. In addition to safety concerns, VANETs can also support other non-safety applications. This includes multimedia (e.g., audio, video) and data services (e.g., internet access, maps, weather) applications.

In a vehicular network at the same way that in any network the concept of routing and its characteristics are highly linked with *Quality of Service (QoS)*. ITU-T (*International Telecommunication Union – T*) Recommendation E.800 [ITU E.800] define QoS as: “*Totally of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service*”. The major concern on VANETs routing is that whether the performance can satisfy the throughput, delay and losses requirements of the applications in a highly mobile environment.

The remainder of project is organized as follows:

- The second chapter introduces an overview of MANETs and VANETs characteristics, applications and services, challenges and research project initiatives.
- The third chapter introduces a background, taxonomy and overview of routing protocols for MANETs that it has been developed or adapted to work in VANETs environments. Besides, it describes QoS challenges in routing for VANETs and some approaches developed which attempt to offer QoS. Also, there is a description of an issue related to compute the available bandwidth on wireless links in order to offer QoS. Finally, there is a description operation of *Ad hoc On-demand Distance Vector (AODV)*, which is a routing protocol widespread use. After that, we described *Available Bandwidth Estimation (ABE)* routing protocol developed in [Sarr 2008] that is able to establish forwarding paths based on bandwidth requirements to provide QoS.
- The fourth chapter describes the capabilities and features, components and architecture, simulation environment and simulation settings of NCTUns simulator [NCTUns] and Citymob mobility pattern generator [Citymob 2008] which are the tools selected in this project to carry out a performance evaluation to compare AODV and ABE.

- The fifth chapter shows the simulations results obtained with the simulator NCTUns and the Citymob mobility pattern generator regarding to packets losses, delay, throughput and packet delivery. There is a comparison between AODV and ABE routing protocol in VANETs environments.
- At last, the sixth chapter gives conclusions and future works that may arise.

1.2 Objectives

The main objective of this project is carry out a performance evaluation of *ABE* routing protocol (*Available Bandwidth Estimation*) [Sarr 2008] and *AODV* routing protocol (*Ad hoc On-demand Distance-Vector*) [Perkins 1999] in *VANETs* (*Vehicular Ad hoc Networks*) urban scenarios to compare the performance for both protocols. With different number of vehicles involved to study the effect and impact that causes the higher number of vehicles in terms of: *packet loss ratio, delay, throughput, throughput over time and packet delivery ratio*.

In order to achieve this main objective, it is necessary to accomplish the following specific objectives:

- Study the state of the art of Vehicular Ad hoc NETWORKS, focusing on routing protocols.
- Study the NCTUns simulator [NCTUns], Citymob [Citymob 2008] mobility pattern generator and AWK language [AWK 2011].
- Define and design the appropriate urban scenarios: vehicles, streets, obstacles (buildings).
- Specify appropriate parameters settings: speeds, transmission range, interference range, fading, path loss and so on.
- Perform the corresponding simulations and analyze the obtained results.

2. INTRODUCTION TO MOBILE AND VEHICULAR AD HOC NETWORKS

The term “*ad hoc*” implies a network established for a special or often extemporaneous service customized to applications. The typical ad hoc network is set up for a limited period of time. Protocols are tuned to particular application, for example send a video stream across the battlefield, find out if a fire started in the forest, establish a videoconference between teams engaged in a rescue mission, among other applications. Those applications may be mobile and the environment may change dynamically. Consequently, ad hoc protocols must self-configure to adjust to environment, traffic and mission changes.

Because of its mobile, non infrastructure nature, an ad hoc network poses new design requirements. The first is self-configuration (of address and routing) in the face of mobility. At the application level, ad hoc network users typically communicate and collaborate (e.g., police, firefighters, medical personnel teams in search and rescue mission). These applications thus require efficient group communication for both data and real time traffic.

2.1 Overview to Mobile Ad hoc Networks (MANETs)

The concept of ad hoc networks is not a new concept. It was introduced by *ARPANET* (*Advanced Research Projects Agency Network*), specifically it can be traced back *DARPA* (*Defense Advanced Research Project Agency*) in the United States in the 70's, when *DoD* (*Department of Defense*), understood the potential of packet switched radio technology to interconnect mobile nodes in a mobile environment without infrastructure. Its development was motivated by the need to provide communication between mobile nodes in the battlefield. For this reason, the goal of *DARPA PRNet* (*Packet Radio Network*) project was to provide packet switched networking to mobile battlefield elements when there is no infrastructure in a hostile environment (soldiers, tanks, aircraft, etc., forming the nodes in the network). Ad hoc networks research stayed long time in the military environment [Ramanathan 2002] [Taneja 2007].

In the early of the 90's a spate of new developments established a new phase, when the notebooks computers became popular, and other communication equipment together with advance of wireless and commercial radio technology became aware of the great advantages of ad hoc networks outside the military battlefield domain. At the same time, the idea of a collection of mobile nodes was proposed at several research conferences and then became in an active research work around ad hoc networks. The IEEE 802.11 sub-committee [IEEE 802.11 WG] formally adopted the term “*ad hoc networks*”, the concept of commercial (non-military) ad hoc networking had arrived and other novel non-military possibilities were suggested, and interest grew.

Powered by the growing interest in ad hoc networking, a lot of hard work were been done on the ad hoc networks standards, evolved in the mid to late 90's. Within the *IETF* (*Internet Engineering Task Force*), the *MANET* (*Mobile Ad hoc NETWORK*) Working Group [MANET WG] was born, with the purpose to standardize *IP* (*Internet Protocol*) functionality suitable for wireless routing application within both static and dynamic topologies with increased dynamics due to node motion and other factors.

2.1.1 Mobile Wireless Networks

The widespread rapid advances of computers and wireless communication stimulated the development of mobile communication systems. There came the need to communicate while on the move, or away from the fixed phone outlet or internet plug. The people became realized to use the technology in the living environments, with the expansion in the field of mobile computing due the proliferation not expensive, available wireless devices such as personal communication like mobile phones, laptops, portable media players, handhelds and so on.

There are currently two variations of mobile wireless networks [Han 2004]: *infrastructure networks* and *infrastructure-less networks*.

2.1.1.1 Infrastructure Wireless Networks

Conventional wireless mobile networks among which it can highlight *WLAN (Wireless Local Area Network)* and the mobile cellular telephony such as: *GSM (Global System for Mobile communications)*, *GPRS (General Packet Radio Services)*, *UMTS (Universal Mobile Telecommunications Systems)*, work under the concept of cells; all of them, require a pre-established fixed infrastructure and centralized mechanisms to carry out their operations and communication processes, default *APs (Access Points)* or *BSs (Base Stations)* are used to allow mobile users to connect the network and obtain the services that require. Besides, the support network topology is static in nature, which facilitates up to a certain point the operational processes of the network.

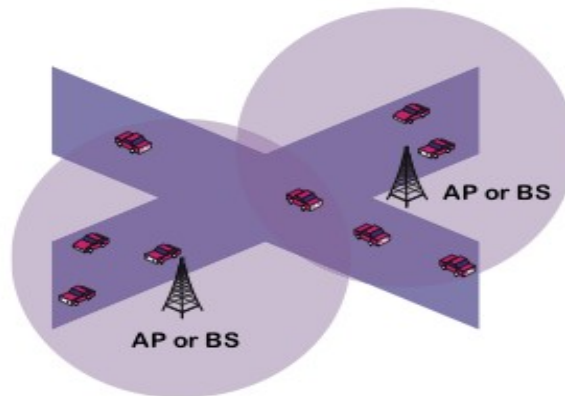


Figure 2.1. Cellular network / WLAN

2.1.1.2 Infrastructure-less Wireless Networks

On the other hand, appear the ad hoc networks that have some different characteristics to those described for previous environments and make a special case of mobile and wireless communications. This type of networks, not need to have some kind of existing physical infrastructure, it does not operate under centralized control schemes. Its topology changes dynamically and randomly, the nodes that constitute an ad hoc network operate as end devices (transmitters or receivers of information) and/or router, basically working in a collaborative environment connectivity.

An ad hoc network, broadly defined is a set of mobile and wireless nodes, which join voluntarily forming a network among themselves, without the need of any centralized administrative entity or existing physical network support in its most basic form.

In simpler words an ad hoc network is self-creating, self-organizing, self-administering.

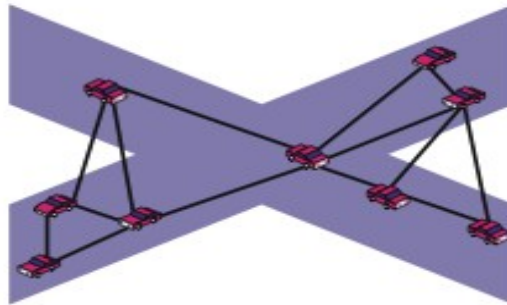


Figure 2.2. Ad hoc network.

2.1.2 Ad hoc Networks Characteristics

Due to its flexibility in deployment, there are many characteristics in ad hoc networks [Gerla 2005]:

- **Mobility.** Is the reason for existence of ad hoc networks, the deployment in areas with no infrastructure implies that the users must form teams that in turn coordinate among themselves to create a task force which can have individual random mobility, group mobility, motion along pre-planned routes, etc. The mobility model have major impact on the selection of a routing scheme and can thus influence performance.
- **Multihopping.** A multihop network is a network where the path from source to destination traverses several other nodes. Ad hoc networks often exhibit multiple hops for obstacle negotiation, spectrum reuse, and energy conservation.
- **Self-organization.** Ad hoc network must autonomously determinate its own configuration parameters including: addressing, routing, clustering, position identification, power control, and so on.
- **Energy conservation.** Most ad hoc nodes have limited power supply and no capability to generate their own power. Energy efficient protocol design (e.g., MAC, routing, resource discovery, etc.) is critical for the duration of the mission.
- **Scalability.** In some applications the ad hoc network can grow to several thousand nodes. For infrastructure wireless networks scalability is simply, its limited mobility can be easily handled using Mobile IP or local handoff techniques. In contrast, because of the more extensive mobility, ad hoc networks do not tolerate mobile IP or a fixed hierarchy structure. In this sense mobility together with the scalability is one of the most critical challenges in ad hoc design.

- **Security.** The challenges of wireless security are well-known, ability of intruders to eavesdrop and jam/spoof the channel. Ad hoc networks are even more vulnerable to attacks than the infrastructure counterparts. Both active and passive attacks are possible, due to the complexity of the ad hoc networks protocols these active attacks are more difficult to detect. Passive attacks can be more insidious than the active ones. The active attacker is eventually discovered and physically disabled/eliminated. The passive attacker is never discovered by the network. Defense from passive attacks requires powerful novel encryption techniques coupled with careful network protocol designs.
- **Unmanned autonomous vehicles.** Some of the popular ad hoc network applications require unmanned, robotic components. All nodes in a generic network are capable of autonomous networking. When autonomous mobility is also added, there arise opportunities for combined networking and motion. For instance, *UAVs (Unmanned Airborne Vehicles)* can cooperate in maintaining a large ground ad hoc network interconnected in spite of physical obstacles, propagation channel irregularities and enemy jamming.
- **Connection to Internet.** For instance, the reach of a domestic wireless LAN can be extended as needed (to the garage, the car parked in the street, the neighbour's home, etc.) with portable routers. These opportunistic extensions are becoming increasingly important and in fact are the most promising pathway to commercial applications. The integration of ad hoc protocols with infrastructure standards is thus becoming a hot issue.

2.1.3 Ad hoc Network Types

The wireless ad hoc network can be divided into two main types [Ali 2010]: *WSNs (Wireless Sensor Networks)* and *MANETs*. In *WSNs* the nodes may be portable or static and *MANETs* the entire network may be mobile and the nodes may move fast relative to each other.

2.1.3.1 Wireless Sensor Networks (WSNs)

A mobile ad hoc sensor network follows an operational sequence and needs less complex setup procedure compared to typical sensor networks, which communicate directly with the centralized controller. A mobile ad hoc sensor or hybrid ad hoc network includes a number of sensors spread in a large geographical area. Each sensor has some level of intelligence to process signals and to transmit data.

In order to support routed communications between two sensor nodes, the routing protocol determines the node connectivity and routes packets accordingly. This condition makes a mobile ad hoc sensor highly flexible so that it can be deployed in almost all environments.

There are many benefits of this network, that:

- Use to build a large scale networks.
- Implement sophisticated protocols.
- Reduce the amount of communication wireless required to perform tasks by distributed and/or local.
- Implement complex power saving modes of operation depending on the environment and the state of the network.

With the aforementioned advances in sensor networking technology, functional applications of WSN increasingly continues surfacing, it requires a low setup and administration costs. Examples include the replacement of existing detecting scheme for forest fires around the world. Using sensor networks, the detecting time can be reduced significantly. Besides, applications in the medical field, home automation and so on. Further information related to applications in WSN is available in [García 2007].

2.1.3.2 Mobile Ad hoc NETWORKS (MANETs)

MANET is a group of independent network mobiles devices that are connected over various wireless links. It works on a constrained bandwidth. Network topologies are dynamic and may vary over time. Each mobile node must act as a router for transmitting the traffic among each other. This network can operate by itself or incorporate into other networks such as *Local Area Networks (LANs)*.

A special kind of MANETs are *Vehicular Ad hoc NETWORKS (VANETs)*. The design of network protocols for such networks faces with multifaceted issues. It needs well-organized distributed algorithms to determine network organization, link scheduling and routing. Conventional routing will not work in this distributed environment because this network topology can change at any point of time. Therefore, it needs some sophisticated routing algorithms that take into consideration this important issue. Some of the factors that have become the core issues include variable wireless link quality, propagated path loss, fading, interface, powered consumed and network topology changes.

2.2 State of the Art in Vehicular Ad hoc Networks (VANETs)

Vehicular Ad hoc Networks are wireless networks that have emerged thanks to advances in wireless technologies and the automotive industry. Vehicular networks are spontaneously formed between moving vehicles equipped with wireless interfaces that could use homogenous or heterogenous technologies. VANETs are considered as one of ad hoc network real-life application enabling communications among nearby vehicles as well as between vehicles and nearby fixed equipment, usually described as roadside equipment.

VANETs in city environments poses many challenges in terms of packet routing. In a city environments there are various obstacles, for instance: high rise buildings and road topologies consisting of many streets and junctions. Apart from having frequent radio interference, packets have to traverse many different paths during the packet forwarding process. These factors have to be considered when designing a new packet forwarding strategy because it would affect the performance of the routing protocol implemented.



Figure 2.3. VANET environment in urban scenario.

VANETs employ variety of advanced wireless technologies such as *DSRC (Dedicated Short Range Communications)* [DSRC], which is an enhanced version of the WiFi technology suitable for VANETs environments. The DSRC is developed to support the data transfer in rapidly changing communication environments, like VANETs, where time-critical responses are required. VANETs are promising in allowing diverse communication services to drivers and passengers. VANETs are receiving considerable attention from the research community as well as the automotive industry.

2.2.1 VANETs Architecture

VANETs are composed of vehicles equipped with *OBU* (*On Board Units*), and stationary nodes called *RSU* (*Road Side Units*) attached to infrastructure that will be deployed along the roads. Both OBUs and RSUs devices have wireless/wired communication capabilities. OBUs communicate with each other and with RSUs in ad hoc manner. There are mainly two types of communications scenarios in vehicular networks: *V2V* (*Vehicle-to-Vehicle*) and *V2R* (*Vehicle-to-RSU*). The RSUs can also communicate with each other and with other networks like Internet, as shown in Figure 2.4.

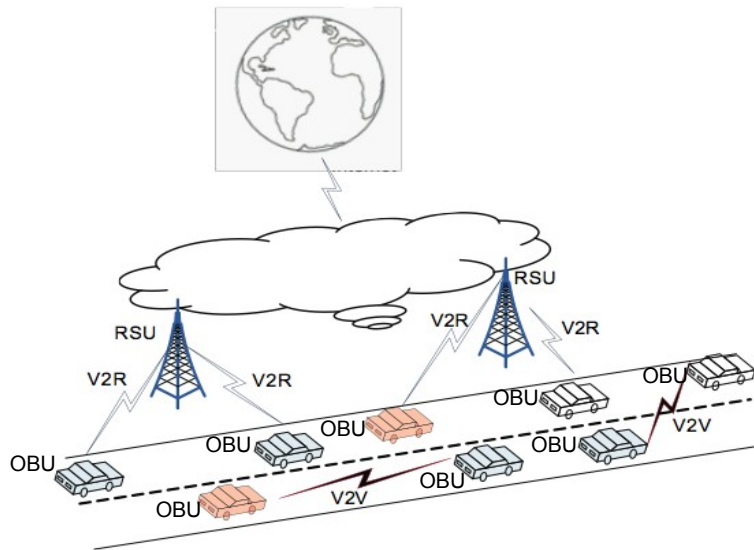


Figure 2.4. VANETs architecture.

2.2.2 VANETs Characteristics

VANETs are characterized by their unique characteristics that distinguish them from MANETs. These special characteristics can be summarized as follows [Yahya 2011]:

- **High mobility.** VANET nodes are characterized by their high relative speed which makes VANET environment high dynamic.
- **Predictable and restricted mobility patterns.** Unlike the random mobility, VANET node movements are governed by restricted rules (traffic flow rules), which make them predictable at least on the short run.
- **Rapid topology change.** VANET nodes are characterized by their high speed. This leads to frequent network topology changes, which introduces high communication overhead for exchanging new topology information.

- *No power constraints.* Each vehicle is equipped with battery that is used as an infinite power supply for all communications and computation tasks.
- *Localization.* Vehicles can use *Global Positioning System* (GPS) to identify their locations with high accuracy.
- *Abundant network nodes.* Unlike MANETs that are characterized by a small network sizes, VANET networks can be very large due to high density of vehicles.
- *Hard delay constraints.* Safety messages are the main goal of VANETs. Therefore, safety messages should be given high priority and must be delivered on time.

2.2.3 VANETs Applications and Services

VANET applications range from road safety applications oriented to vehicle or to driver, to entertainment and commercial applications for passenger, making use of a big quantity of cooperating technologies. The main goal of VANETs include real-time and safety applications for drivers and passengers, providing safety by giving essential tools to decide the best route along the road. These applications seek to minimize accidents and improve traffic conditions by providing drivers and passengers useful information including collision warnings, road signal alarms and so on.

VANETs promote useful driver and passenger oriented services, which include Internet connections facility exploiting available infrastructure on-demand way, electronic tolling system and multimedia services. As well as, communication networks such as 2-3G, WLANs IEEE.11a/b/g/p and WiMAX, can be exploit to enable new services designed for passengers from part of safety applications, such as info-mobility and entertainment applications. Table 2.1 shows some examples of applications and its requirements.

These applications are categorized as safety and non-safety applications and they have different requirements. Safety messages should be given higher priority over the non-safety messages. Safety messages are time-sensitive and should be disseminated to vehicles in the surrounding area of the event within a bounded time. Safety messages are either event-driven or periodic based. For instance, event-driven safety messages are high priority messages generated and sent by vehicles involved in an accident to warn vehicles approaching the accident area [Moustafa 2008] [Yahya 2011].

<i>Applications</i>	<i>Packet Size (bytes)/Bandwidth</i>	<i>Delay (ms)</i>	<i>Network Data Type</i>	<i>Application Range (m)</i>	<i>Priority</i>
<i>Intersection collision warning / avoidance</i>	~ 100	~ 100	Event	300	Safety of life
<i>Cooperative collision warning</i>	~ 100/10 Kbps	~ 100	Periodic	50 - 300	Safety of life
<i>Work zone warning</i>	~ 100/1 Kbps	~ 1000	Periodic	300	Safety
<i>Transit vehicle signal priority</i>	~ 100	~ 1000	Event	300 - 1000	Safety
<i>Tolls collections</i>	~ 100	~ 50	Event	15	Non-safety
<i>Services Announcements</i>	~ 100/2 Kbps	~ 500	Periodic	0 - 90	Non-safety
<i>Movie Download (2 hrs of MPEG 1)</i>	> 20 Mbps	NA	NA	0 - 90	Non-safety

Table 2.1. Examples of applications and requirements.

On the other hand, periodic safety messages are considered preventive safety methods sent at specific intervals. Periodic messages carry the current status like velocity, acceleration, direction, etc. These information is used by vehicles on the neighbourhood to update the status of their neighbourhood. Non-safety applications have different goals and can be used to provide a number of services ranging from transportation management, toll collections, infotainment, music download, commercial advertisements and so on. Non-safety data should be given low priority compared to the safety data.

Regarding the applications, VANETs open new business opportunities for car manufactures, automotive, network operators, services providers and integrated operator in terms of infrastructure deployment as well as service provision and commercialization. For safety applications, the network operator can assure the authentication of each vehicle by playing the role of trusted third party that authenticates the participating vehicles, or even having the role of a certifications authority issuing a certificate to each participant in order to prove the authenticity of them during the communication. On the other hand, in non-safety applications, network operator and/or service providers, besides network access and services provision, can have a role of authorizing services access and billing users for the consumed services.

2.2.4 Challenges in VANETs

The above unique characteristics create new challenges that need to be resolved in the vehicular network environments. The main challenges in VANETs can be summarized as follows [Moustafa 2008]:

- **Potentially large scale.** Unlike most ad hoc networks, usually assume a limited network size, VANETs can in principle extend over the entire road network and so include many participants.
- **High mobility.** The environment in which VANETs operate is extremely dynamic, and includes extreme configurations; on highways, relative speeds of up to 150 km/h may occur, while density of nodes may be 1 or 2 vehicles per 1 km on low busy roads. On the other hand, in the city relative speeds can reach up to 60 km/h and vehicles density can be very high, especially during rush hour.
- **Partitioned network.** VANETs will be frequently partitioned. The dynamic nature of traffic may result in large inter-vehicle gaps in sparsely populated scenarios, and hence in several isolated clusters of nodes.
- **Network topology and connectivity.** VANET scenarios are very different from classic ad hoc networks. Since vehicles move and change their position constantly, scenarios are very dynamic. Therefore, the network topology changes frequently as the links between vehicles connect and disconnect very often. The degree to which the network is connected is highly dependent on two factors: the range of wireless links and the fraction of participant vehicles, where only a fraction of vehicles on the road could be equipped with wireless interfaces.

2.2.4.1 Technical Challenges

A number of technical challenges need to be resolved in order to deploy VANETs and provide useful services for drivers and passengers. Scalability and interoperability are two issues that should be satisfied, and the employed protocols and mechanisms should be scalable to numerous vehicles and interoperable with different wireless technologies. This technical challenges are regarding to [Moustafa 2008]: *Reliable Communications and MAC Protocols, Routing and Dissemination, Security, IP Configuration and Mobility Management, Application Distribution and Business Models*.

2.2.4.1.1 Reliable Communications and MAC Protocols

Multihop wireless communication represents a major challenge on the reliability of communication. In consequence, efficient MAC access protocols need to be in place, while adapting the highly dynamic environment of VANETs, and considering messages priority of some applications (e.g., accident warnings). In spite of the dynamic topology and the high mobility, low communication latency should be satisfied between communicating vehicles in order to guarantee service reliability for safety applications by taking into consideration the time-sensitive during message transfer and the continuity of service for non-safety applications. Moreover, MAC protocols should take into account the heterogenous communication between different wireless technologies (e.g., WiFi, GSM).

2.2.4.1.2 Routing and Dissemination

VANETs differ from conventional ad hoc wireless networks not only experiencing rapid changes in wireless link connections, but also having to deal with different types of network densities. For instance, urban areas are more probably to form highly dense networks during rush hour traffic and freeways are expected to experience frequent network fragmentation in sparsely rural freeways or during late night hours. In consequence, routing and dissemination algorithms should be efficient and should adapt to VANET characteristics and applications, permitting different transmission priorities according to the application type safety or non-safety. For instance, message dissemination in safety applications should be broadcast, in a way to assure the message propagation to the required cluster of vehicles without causing a broadcast storm. In non-safety applications, message transfer through unicast or multicast transmission is more suitable.

2.2.4.1.3 Security

Security and privacy are major concerns in the development and acceptance of services and should not be compromised by ease of use of service discovery protocols. As the demand for service discovery is growing, passengers may use services in foreign networks and create immense security problems for themselves and for other network users. In consequence, it is important to propose innovative solutions for secure communications between participants as well as authorized and secure services access. Appropriate security architectures should be in place providing communication between vehicles and allowing different service access. As well as, a set of security mechanisms suitable for any VANET environment should be developed, providing trust, authentication, access control and authorized and secure service access.

2.2.4.1.4 IP Configuration and Mobility Management

The potential *V2R (Vehicle-to-RSU)* architecture is promising in allowing vehicular internet access as well as provision of Internet related services to drivers and passengers. However, two challenges exist under this issue: IP address configuration and mobility management. These challenges can threaten the service quality and the service continuity. IP address configuration should be carried out in an automatic and distributed manner. There is no standard for IP auto-configuration in ad hoc networks, and hence the problem becomes complex for VANETs. For mobility management, this is a crucial problem for non-safety applications, where messages dissemination is not broadcast. The absence of mobility management mechanisms threatens service commercialization and loses the benefit of *V2R* architecture since internet services neither guarantee service quality nor their continuity.

2.2.4.1.5 Business Models

Business models represent an important challenge for service commercialization in VANETs. As a matter of opening a new business opportunity, business models should be rentable for telecom operator and service providers aiming to promoting services and attracting clients. In consequence, special accounting mechanisms and tailored billing systems are needed, which also assure inter-domain accounting. However, processing delay and integrity, where the operator could assure the authentication and secure communication between clients in a way that protects the clients' data and allows for billing the used services.

2.3 Standardization and Research Projects Initiatives

High interest for VANETs is also from governmental authorities and standardization organizations. In this context, *DSRC (Dedicated Short-Range Communications)* systems has emerged in 2003. On the other hand, *CAR-2-CAR Communication Consortium* has been initiated in Europe by car manufactures with the main objective of increasing road traffic safety and efficiency by means of inter-vehicle communication. IEEE is also advancing within the IEEE 1609 family of standards for *WAVE (Wireless Access in Vehicular Environments)*. Different projects and research groups contribute to develop this kind of networks, some of them are presented below.

2.3.1 Dedicated Short-Range Communications (DSRC) Technology

DSRC technology is used in the ITS domain to provide secure and reliable communication links *V2V (Vehicle-to-Vehicle)* and *V2R (Vehicle-to-RSU)*. These communication links allow the transfer of data that are necessary for the operation of different ITS applications. DSRC is developed to work in very high dynamic networks to support fast link establishments and to minimize communication latency. Mainly, the DSRC is designed to ensure the service reliability for safety applications taking into account the time constraint for this type of applications and it can also support non-safety applications [DSRC].

2.3.2 CAR-2-CAR Communication Consortium

The mission and the objectives of the CAR-2-CAR Communication Consortium are [CAR-2-CAR]:

- The development and release of an open European standard for cooperative *ITS (Intelligent Transportation Systems)* and associated validation process with focus on inter-vehicle communication systems.

- To be a key contributor to development of a European standard associated validation process for V2R infrastructure communication being interoperable with the specified inter-vehicle communication standard.
- To provide its specifications and contributions to standardization organizations including particular *ETSI (European Telecommunications Standards Institute) TC (Technical Committee) ITS* in order to achieve common European standards for ITS.
- To push harmonization of CAR-2-CAR communication standards worldwide.
- To promote allocation of a royalty free European wide exclusive frequency band for CAR-2- CAR applications.
- To develop realistic deployment strategies and business models to speed up the market penetration.
- To demonstrate CAR-2-CAR system as proof of technical and commercial feasibility.

2.3.3 WAVE (Wireless Access in Vehicular Environments) stack

Efforts have been made to design new standards for the services and the interfaces for VANETs. These standards form the basis for wide range of applications in VANETs environments. A trial of a set of standardized services and interfaces defined under WAVE stack has been released. These services and interfaces enable a secure V2V and V2R communications in a changing communications environment, where communications and transactions need to be completed in a short time.

The WAVE architecture is developed based on the IEEE 802.11p and the IEEE P1609 standards. IEEE 802.11p [IEEE 802.11p] deals with the physical and *MAC (Media Access Control)* layers, whereas IEEE P1609 [IEEE P1609] deals with the higher-layer protocols. Figure 2.5 shows IEEE WAVE stack. For further information about IEEE WAVE is available in [IEEE P1609] and IEEE 802.11p is available in [IEEE 802.11p].

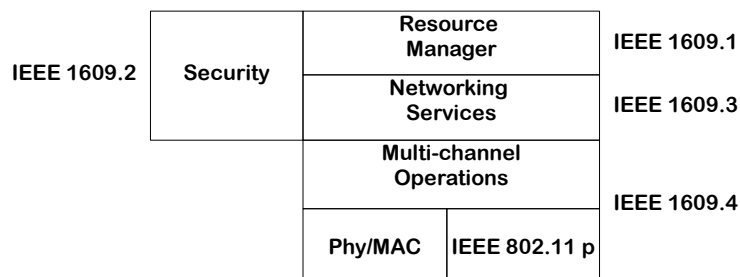


Figure 2.5. IEEE WAVE stack.

- *IEEE 1609. 1: Resource Manager.* Defines the services and the interfaces of WAVE Resource Manager applications. It describes the message formats and the response to those messages. It also describes data storage format that is used by applications to access other architectures.
- *IEEE 1609. 2: Security Services.* Defines security and secure message formatting and processing. It also defines how secure messages are exchanged.
- *IEEE 1609. 3: Networking Services.* Defines routing and transport layer services. It also defines a WAVE specific message alternative that can be supported by the applications. This standard also defines the *Management Information Base (MIB)* for the protocol stack.
- *IEEE 1609. 4: Multi-channel Operations.* Defines the specifications of the multi-channel in the DSRC. This is basically an enhancement to IEEE 802.11a *Media Access Control (MAC)* standard.

2.3.4 CARLINK

The aim of this project is to develop an intelligent wireless traffic service platform between vehicles supported by wireless transceivers beside the road. The primary applications are real-time local weather data, urban transport traffic management, and urban information broadcasting. Vehicles have integrated wireless transceivers to communicate with base stations located beside the road. In addition, vehicles may also communicate between each other as members of an ad hoc network. Base stations provide real-time information (e.g., weather, traffic density) to vehicles driving past. At the same time, vehicles gather real-time data and deliver this information back to base stations. Base stations provide all gathered information to a central unit, which updates its databases and provides information on current traffic conditions, weather, and so on [CARLINK].

2.3.5 PReVENT

The three year WILLWARN subproject is developing, integrating and validating a safety application that warns the driver whenever a safety related critical situation occurring beyond the driver's field of view. This includes the development of on board hazard detection, in car warning management, and decentralized warning distribution by V2V communication on a road network. Positioning, relevance checks, message transport, and on board message evaluation will enable a low cost and reliable solution for wireless local danger warnings [PReVENT].

2.3.6 Network on Wheels (NOW)

Network on Wheels was funded by Daimler AG, BMW AG, Volkswagen AG, Fraunhofer Institute for Open Communication Systems, NEC Deutschland GmbH and Siemens AG in 2004. The main objectives are to solve technical key questions on the communication protocols and data security for V2V

communications and to submit the results to standardization activities of CAR-2-CAR Communication Consortium. Furthermore, a test bed for functional tests and demonstrations is implemented which will be developed further on toward a reference system for the CAR-2-CAR Communication Consortium specifications [NOW].

2.3.7 SAFESPOT

SAFESPOT aims to [SAFESPOT]:

- Use infrastructure and vehicles as sources and destinations of safety-related information and develop an open, flexible and modular architecture and communication platform.
- Develop the key enabling technologies: ad hoc dynamic network, accurate relative localization, dynamical local traffic maps.
- Develop and test scenario-based applications to evaluate the impacts on road safety.
- Define a sustainable deployment strategy for cooperative systems for road safety, evaluating also related liability, regulations and standardization aspects.

2.3.8 Cooperative Vehicle-Infrastructure Systems (CVIS)

The CVIS objectives are [CVIS]:

- To create a unified technical solution allowing all vehicles and infrastructure elements to communicate with each other in a continuous and transparent way using a variety of media and with enhanced localization.
- To enable a wide range of potential cooperative services to run on an open application framework in the vehicle and roadside equipment.
- To define and validate an open architecture and system concept for a number of cooperative system applications, and develop common core components to support cooperation models in real-life applications and services for drivers, operators, industry and other key stakeholders.
- To address issues such as user acceptance, data privacy and security, system openness and interoperability, risk and liability, public policy needs, cost/benefit and business models, and roll-out plans for implementation.

2.3.9 Privacy Enabled Capability In Co-Operative Systems and Safety Applications (PRECIOSA)

The goal of PRECIOSA is to demonstrate that co-operative systems can comply with future privacy regulations by demonstrating that an example application can be endowed with technologies for suitable privacy protection of the location related data of individuals.

The major objectives of the PRECIOSA project are to [PRECIOSA]:

- Define an approach for the privacy evaluation of co-operative systems in terms of communication privacy and data storage privacy.
- Define a privacy aware architecture for co-operative systems which involves suitable trust models and ontologies, a V2V privacy verifiable architecture, and a V2R privacy verifiable architecture, and which includes the architecture components for protection, infringement detection and auditing.
- Define and validate guidelines for privacy aware co-operative systems.
- Investigate specific challenges for privacy.

2.3.10 Secure Vehicle Communication (SEVECOM)

The following research and innovation work is foreseen [SEVECOM]:

- Identification of the variety of threats: attacker's model and potential vulnerabilities; in particular, study of attacks against the radio channel and transferred data, but also against the vehicle itself through internal attacks.
- Specification of architecture and security mechanisms which provide the right level of protection. It will address issues such as contradiction between liability and privacy, or the extent to which a vehicle can check the consistency of claims made by other vehicles. The following topics will be fully addressed: key and identity management, secure communication protocols, tamper proof device and decision on crypto-system, privacy, intrusion detection, data consistency, secure positioning, secure user interface.

In next chapter, we introduce a background, taxonomy and overview of routing protocols for MANETs that have been developed or adapted to work in VANET environments. Besides, it describes QoS challenges in routing for VANETs and some approaches developed which attempt to offer QoS. Also, there is a description of an issue related to compute the available bandwidth on wireless links in order to offer QoS. Finally, there is a description operation of *Ad hoc On-demand Distance Vector (AODV)*, which is a routing protocol widespread used. After that, we described *Available Bandwidth Estimation (ABE)* routing protocol developed in [Sarr 2008] that is able to establish forwarding paths based on bandwidth requirements to provide QoS.

3. ROUTING PROTOCOLS FOR VEHICULAR AD HOC NETWORKS (VANETs)

Routing is the process of forwarding data from a source to its destination, sometime requiring multihop forwarding nodes. Specifically, routing protocols are responsible of determining the paths to forward the packets to their destination, and also find alternative paths in case of failure. In general, an efficient routing protocol is one that is able to deliver packets in a short amount of time and consuming minimal bandwidth.

3.1 Background Routing for VANETs

Differently to the routing protocols implemented in MANETs, routing protocols for VANET environments must take into account the main issue: the highly dynamic topology. VANETs are formed with vehicles joining and leaving the network all the time and sometimes only stay for a few seconds.

Applying traditional MANET routing protocols in VANETs environments is inefficient, since these methods do not take the VANETs characteristics into account, principally the fast mobility. Therefore, modifying MANET routing protocols or developing new routing protocols specific for VANETs are the usual approaches to efficiently implement routing protocols in VANETs.

3.1.1 Taxonomy of Routing Protocols

This section presents a general overview of the taxonomy of new VANET routing protocols or other that have been adapted for VANETs. They are classified into 5 categories, as shown in Figure 3.1, according to the type of information in which they trust to take the routing decisions. Nevertheless, in the literature exists another classification according to data disseminations in [Wei Lin 2010].

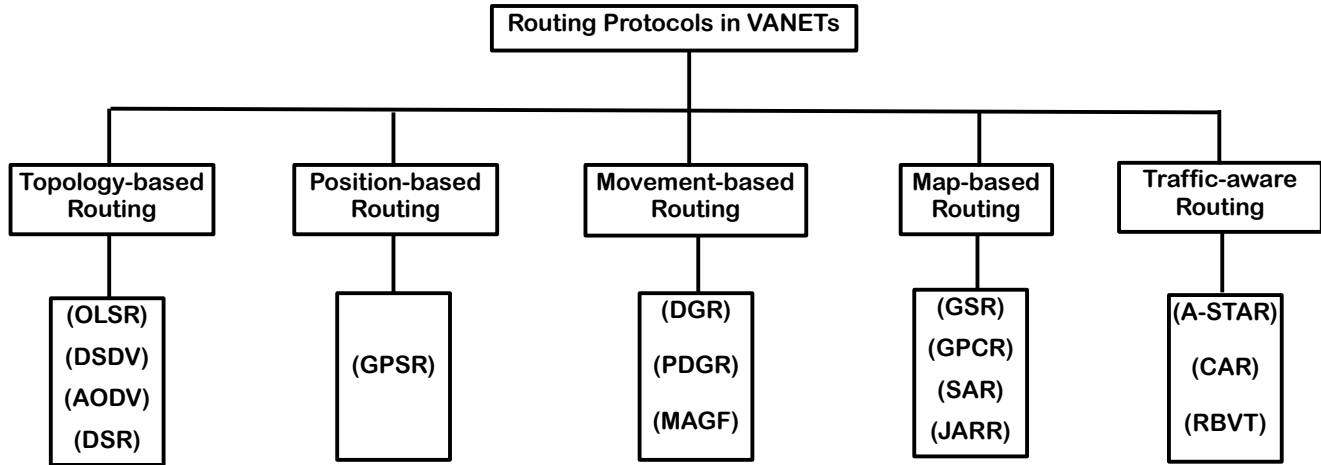


Figure 3.1. Taxonomy of routing protocols in VANETs.

3.1.1.1 Topology-based Routing

Topology-based routing protocols attempt to balance between being aware of all possible paths and keeping the overhead at the minimum level. These protocols use information about network topology and the state of communication links between nodes to take the routing decisions. Those that keep an information table about your neighbours are called proactive protocols instead, reactive protocols forward a packet on the fly.

Proactive protocols build their routing tables based on the current connectivity of the information nodes. They continuously attempts to keep up to date their information, and they are developed to work in low mobility environments. However, [Benzaid 2002] proposed a *fast OLSR (Optimized Link State Routing)* where vehicles exchange topology information using beacons to build routing paths, the exchange of beacon messages is optimized and adapted to the network dynamics.

In *DSDV (Destination-Sequenced Distance-Vector Routing)* [Perkins 1994], every node maintains a distance vector to every destination. Therefore, frequent broadcast messages are sent by all nodes to

learn and advertize topology changes. In consequence, it consumes a considerable amount of bandwidth, this is due to a large amount of data that exchanged for routing maintenance, especially in high dynamic networks where the nodes neighbourhood is always changing, which increase the overhead needed to maintain routing tables.

That additional traffic used in proactive approaches for maintenance of unused paths has several drawbacks: it consumes network resources, it wastes bandwidth for control messages, and usage of flooding increase network congestion. There is a tradeoff between the routing information freshness and the control overhead. In conclusion, proactive solutions do not scale well in large networks with a high number of vehicles joining and leaving the network by a short time, which is the case in VANETs.

Reactive protocols depend on flooding the network with query packets to find path to destination. They determine a path to a given destination only on demand. *AODV (Ad hoc On-demand Distance Vector)* [Perkins 1999] and *DSR (Dynamic Source Routing)* [Johnson 1996], are reactive topology-based routing protocols. They reduce the overhead by restricting route maintenance only between nodes that need to communicate. In AODV, a node floods a query packets that are forwarded until reach their destination. Each node along the path towards the destination adds its address to list of relay nodes carried in the packet. When the destination is reached, its responds to the source listing the path taken and intermediate nodes store this connectivity information. AODV is described more into detail in section 3.3.

3.1.1.2 *Position-based Routing*

Also called Geographic-based routing protocols they trust on the current vehicles locations to determine the path to forward a packet. These protocols take the routing decisions based on the geographic vehicles information. They assume that vehicles are equipped with *GPS (Global Position System)*. Therefore, they require that all vehicles know their physical positions as well as their neighbours. They also assume that source vehicles know the position of the destination.

One example of these protocols is *GPSR (Greedy Perimeter Stateless Routing)* [Karp 2000], which has several requirements on the availability of position information: GPSR requires that each vehicle is able to obtain its current location, it assumes that each vehicle learns about the existence of its direct neighbours and their current positions through the exchange of periodic hello messages. To take a routing decision, a source vehicle needs to know the position of the destination and it forwards the packets to its neighbour which is geographically closest to the destination. This procedure, known as *Greedy Forwarding* is recursively applied by intermediate vehicles until final destination is reached.

However, it can occur that the packets reach vehicles that has not neighbour closer to the destination. This lead to a local maximum problem that it appears when there is not vehicle to which forward a packet.

3.1.1.3 Movement-based Routing

Due the local maximum problem, *DGR (Directional Greedy Forwarding)* [Gong 2007] and *MAGF (Movement Aware Greedy Forwarding)* [Brahmi 2009] address this shortcoming. Both protocols make use of additional information about vehicles. The main idea is to compute a score as a function of factors such as: position, direction, and speed. In order to assign priority between neighbours vehicles while select the next forwarder. This enhancement of position-based routing reduces the number of local maximum problem by avoiding sending packets from the destination by selecting a wrong direction.

Considering that vehicles follow a predictable mobility pattern, for instance in highway scenarios, the authors of DGR proposed *PDGR (Predictive Directional Greedy Routing)* [Gong 2007] to forward a packet to the most suitable next hop based on both current and predictable future locations.

3.1.1.4 Map-based Routing

Map-based routing protocols combine geographic information with topological knowledge. In [Lochert 2003], it has been proposed *GSR (Geographic Source Routing)* which depends on the cities maps and the locations of the source and destination vehicles.

In GSR streets intersections can be seen as junctions, which represents the *path* that the packets have to cross to reach their destinations. GSR uses the greedy forwarding technique (it makes a local optimal choice at each stage) to determine the next junction location on the path. A packet that is received is forwarded to the vehicle that is closer to the next junction. This process is repeated until the packet is delivered to its final destination. There are two ways to learn junctions sequence: the first requires the whole junctions list is included in the packet header: the computation complexity and overhead is reduced, but bandwidth usage is increased. The second requires that each vehicle computes the junctions list: the bandwidth consumption is reduced, but computation and overhead increases. There is an improvement for this protocol named *GPCR (Greedy Perimeter Coordinator Routing)* [Lochert 2005].

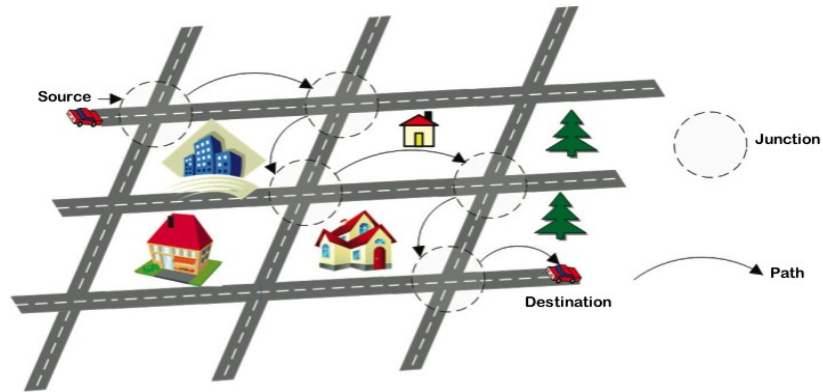


Figure 3.2. Junctions and paths to forward packets in GSR, SAR and JARR.

SAR (Spatially Aware Routing) [Tian 2003] is similar to GSR since packets are forwarded to the next junction until they are delivered. SAR uses three different ways to learn when the forwarding vehicle can not find another vehicle closer to the next junction on the path. The first option is storing the packet and periodically trying to forward it, and it will be discarded until the time limit is spent or the buffer becomes full. The second option is forward the packet, using the traditional greedy forwarding routing, toward the destination instead of to the next junction. Third option is recalculate a new path based on the current situation.

JARR (Junction Based Adaptive Reactive Routing) [Tee 2010] uses different algorithms when the packet is forwarded to a junction, and when it has reached a junction. First, packet is forwarded using an optimal path to a junction. At that point different algorithms determine the next optimal path and alternative paths. JARR takes into account speed, direction, current position, and density when determining the path for a packet. In order to gather this information, a beacon regularly informs about position and speed. JARR is able to exploit the benefits of beacon without paying the full price in overhead by adapting the frequency of beacon as vehicle density increase. The higher the density, the less frequently the beacon is used to disseminate information.

3.1.1.5 Traffic-aware Routing

Traffic-aware routing protocols use available data about vehicular traffic density, and only streets where vehicles are moving will be used to forwarding packets.

In **A-STAR (Anchor Based Street and Traffic Aware Routing)** [Seet 2004], the packet is forwarded along a directional vector that contains anchors or fixed geographic points that the packet must go through. It calculates the best path based on the traffic information. It selects streets with higher vehicle density,

because it provides better transmission and less delay for a packet traveling along it. This traffic information can be determined by the number of bus stops on a street, or by real time measurement of traffic density. The first method is called *statistically rated map* and the second method is called *dynamically rated map*. When a path fails, the anchor path is recalculated and vehicles are notified that this particular path is out of service.

CAR (Connectivity Aware Routing protocol) [Yang 2008] uses statistical data collected by different vehicles to estimate the probability of connectivity of each road segment. It also considers the clustering phenomenon as a result of vehicles movement affected by traffic lights. However, there is a high overhead generated by collecting and exchanging connectivity information about the entire network especially that this information is volatile due to vehicles mobility.

RBVT (Road-Based with Vehicular Traffic protocols) [Nzouonta 2009] protocols incorporate real time vehicular traffic to compute road based paths consisting of successions of road intersections connected among them through vehicular communications. Two variants of RBVT are presented: reactive protocol RBVT-R and proactive protocol RBVT-P. In RBVT-R only source vehicles discover the connected road segments on demand by initiating route discovery packets which traverse the network towards the destination. RBVT-P maintains a graph of all connected road segments to discover the network topology. *Connectivity Packets (CP)* are generated periodically by multiple vehicles randomly selected in the network. That improvement is due real traffic takes into account take the routing decisions adapted to the network conditions. Nevertheless, this procedure generates an additional overhead to maintain the freshness of the topology information.

All these current routing protocols for VANETs described in this section have difficulties to provide certain level of *QoS (Quality of Service)*. A QoS-aware routing protocol should guarantee satisfactorily a certain level of performance. On infrastructure wireless networks this often is achieved through resource reservation and dedicated infrastructure [IEEE 802.11e]. Nevertheless, infrastructure-less nature dynamic and random movement pattern in VANETs environments makes it difficult to do a resource reservation. QoS provisioning is a challenge in VANETs due to their special features which result in frequent routing path breakages.

3.2 Quality of Service Routing for VANETs

The most relevant issue to provide QoS in VANETs is the intermittent connectivity of vehicles caused by high dynamic mobility. The high mobility of vehicles joining and leaving the network all the time makes the connectivity among them very unstable, so even the best effort service can not be guaranteed. There are several applications that generate multimedia data flows or are based on the correct and efficient transmission of susceptible control traffic. These applications may benefit from a QoS support in the network. However, the term QoS is quite ambiguous and involve several concepts.

VANETs have been envisioned with three types of services in mind: *safety*, *traffic management* and *commercial applications*. Vehicles will be able to inform other vehicles about traffic accidents, hazardous road conditions and traffic congestion. Commercial applications for instance, data exchange, infotainment, audio and video communication are another reasons envisioned to provide faster adoption of the technology. However, some services are critical, such as crash warning, and require low delay; other services need bandwidth, such as video-streaming applications.

A new proposed routing protocol, which aims to be suited to a mobile environment, must be based on selecting the best path among all possible choices and also must try to offer QoS to the applications based on the transmission characteristics for instance: bandwidth, delay, packet loss or network load. To achieve this, it is necessary an accurate evaluation of conditions of the paths. Most of the current QoS proposals leave this problem aside, because they assume that the link layer protocol is able to perform such an evaluation. The resource evaluation problem must take into account several phenomenon related to the wireless environment but also depend on less measurable parameter such as vehicle mobility.

There are some algorithms to improve QoS performance in vehicular routing that have been proposed. [Zhaomin 2006] uses a metric called *Expected Disconnection Degree (EDD)* to evaluate the quality of each possible path between source and destination. The value of EDDs reflects the probability that the path would be broken in a certain time period, and is determined by the information of the predicted velocity and moving trajectory of each vehicle along the path. In [Yan 2010] the proposed scheme disseminates packets among the links which has longer expiration time calculated using relative velocity vectors. The optimal routing path and a backup routing path are selected by balancing stability, cost and delay. [Rontala 2011] proposes a scheme which focuses on identifying optimal paths using the idea of next hop selection considering metrics to achieve optimal QoS such as delay, packet loss and bandwidth.

3.2.1 Available Bandwidth (AB)

In order to offer services over multihop ad hoc networks with good performance, QoS mechanisms often require an estimation of available bandwidth and it is a difficult task. The shared nature of the wireless channel and interference between wireless nodes make evaluating *Available Bandwidth (AB)*. Estimating the remaining bandwidth at a given time and in a given part of the network is a hard task because, in a wireless network the medium is shared between close nodes. There are principally three categories that has been adopted to estimate AB.

The first category is measurement-based on *Probe Rate Models (PRM)* and *Probe GAP Models (PGM)*. In [Strauss 2003], PRM is based on the concept of self induced congestion, it sends probe traffic at a rate lower than the available bandwidth along the path, than the arrival rate of probe traffic at the receiver will match their rate at the sender; PGM exploits the information in the time gap between the arrivals of two successive probes at the receiver. The main drawback of these approaches is that it includes high control overhead.

The second category is based on mathematical models such as a Markov model [Bianchi 2000]. It is concentrated on calculating the maximum and saturation throughput performance. The main drawback is that it is topology dependent and in high mobility environments with a random topology, it would be difficult to obtain and maintain the information required by an analytical model.

The third category is calculation-based on effective capacity [Wu 2003], that constructs certain performance metric as a local information on the used bandwidth, to evaluate AB. This local information is broadcasted by hello messages that are implemented in many routing protocols to discover local topology. Regarding the calculation-based category there are several approaches:

In [Chaudet 2001] it has been proposed *BRuIT (Bandwidth Reservation under InTerferences Influence)*. It provides to the nodes information about their neighbours, each node periodically broadcasts a hello message to every other node that can hear it (e.g., that is in its communication range). This hello message contains the address of the transmitter and the total bandwidth that it will use to forward the already accepted flows. Upon reception of such hello message, a node can compute the remaining bandwidth it can use for new flows. Therefore, admission control process decides if a new request is accepted or refused.

[Xu 2003] includes adaptive bandwidth management to measure available bandwidth per node. This bandwidth information is propagated proactively or retrieved on demand. Source nodes perform call admission control based on the bandwidth information provided, even though admission control is performed to guarantee enough available bandwidth before accepting any flow, the network can still experience congestion due to mobility.

In [Zhai 2004] it has been proposed *CARC (Call Admission and Rate Control)*. It proposes two algorithms that function upon the use of the channel busyness ratio; the call admission control algorithm is used to regulate the admission of real-time traffic or streaming traffic and, rate control algorithm to control the transmission rate of best effort traffic.

In [Renesse 2004] it has been proposed QoS-AODV. Its evaluation mechanism constantly updates a value called *Bandwidth Efficiency Ratio (BWER)*, which is the ratio between the number of transmitted and received packets. The available bandwidth is calculated by multiplying the BWER value by the channel capacity. The available bandwidth to a node is then inferred from these values as the minimum of the available bandwidths.

In [Yang 2005] it has been proposed *CACP (Contention-aware Admission Control Protocol)*. It broadcasts a route request that contains the bandwidth requirement. It records the sequence of hops taken by the route request as it is propagated through the network. This sequence of hops is called partial route. Therefore, when a node receives a route request, partial admission control is performed by comparing available bandwidth with the possibly underestimated bandwidth consumption that is calculated by partial route.

In [Wu 2007] it has been proposed *FAT (Fraction of Air Time)*. It makes an estimation by representing the normalized utilization or available capacity resource in two forms: consumed and residual. The consumed/residual fraction of air time is defined as the ratio of the total air time (cost) consumed/available in a given time interval to the length of the interval. The length of the interval should be sufficiently large relative to the air time cost of a packet.

In [Rennesse 2007] it has been proposed *ACC (Adaptive Admission Control)*. It makes use of hello messages forwarded between nodes for connectivity awareness to propagate aggregated bandwidth information.

In [Zhao 2009] it has been proposed *IAB (Improved Available Bandwidth)*. It takes the synchronization between sender and receiver into consideration. It also considers the dependence of the channel idle time sensed by sender and receiver via differentiating the nodes busy and sense busy states.

In [Van Nam 2011] it has been proposed *RABE (Retransmission-based Available Bandwidth Estimation)*. It integrates the impact of retransmission attempts on the available bandwidth. It takes into account, in its estimation, the bandwidth wasted by extra waiting times and medium occupancy due to retransmissions. This estimation requires to compute the collision probability and the mean number of retransmission attempts.

As we can see, there are several routing protocols proposed for VANETs and as it was mentioned at the beginning of this chapter one option that has been adopted is modifying MANETs protocols to work in the VANETs environments or another option is to develop new ones. The authors in [Sarr 2008] design a routing protocol able to establish forwarding paths that fulfill bandwidth requirements and surely to provide certain level of QoS. To this end, the authors study the possibility to use, in realistic VANET urban scenarios, the *Available Bandwidth Estimation (ABE)* that was proposed and evaluated for MANETS. And the main objective of this project is to do a performance evaluation of ABE in VANET urban scenarios with different number of vehicles and mobility patterns to study the impact in terms of packet loss, delay, throughput and packet delivery.

ABE is used to assist AODV routing protocol so that the established forwarding path satisfies the bandwidth required by the application. That is, ABE uses the same basic operation than AODV to find paths and includes an algorithm to improve the performance of QoS parameter. This extension is named hereafter AODV-ABE.

3.3 *Ad hoc On-demand Distance Vector (AODV)*

Ad hoc On-demand Distance-Vector (AODV) is a reactive unicast routing protocol especially designed for MANETs. It was jointly developed in Nokia Research Center and in University of California and University of Cincinnati by C. Perkins, E. Belding-Royer and S. Das [RFC 3561].

As a reactive routing protocol, it maintains routing information about the active paths, every node keeps a next hop routing table, which contains the destinations towards which currently has a path. A routing table entry expires if it has not been used or reactivated for an specific expiration time. Moreover, AODV adopts destination sequence number technique.

3.3.1 AODV Operation

AODV in accordance with [RFC 3561] performs two main tasks: *Route Discovery* and *Route Maintenance*.

3.3.1.1 AODV Route Discovery

When a source node wants to forward packets to a destination but there is not available path, it start a route discovery operation (see Figure 3.3) which consists in the source broadcasts *Route Request (RREQ)* packets. These RREQ includes the addresses from source and destination; the broadcast ID, which is used as identifier; the last seen sequence number of destination as well as the sequence number of the source node. The sequence numbers are important to ensure loop-free and up to date paths. To reduce the flooding overhead, a node discards RREQs that it has seen before. The RREQ starts with small *TTL (Time To Life)* value; if the destination is not found, TTL is increased in following RREQs.

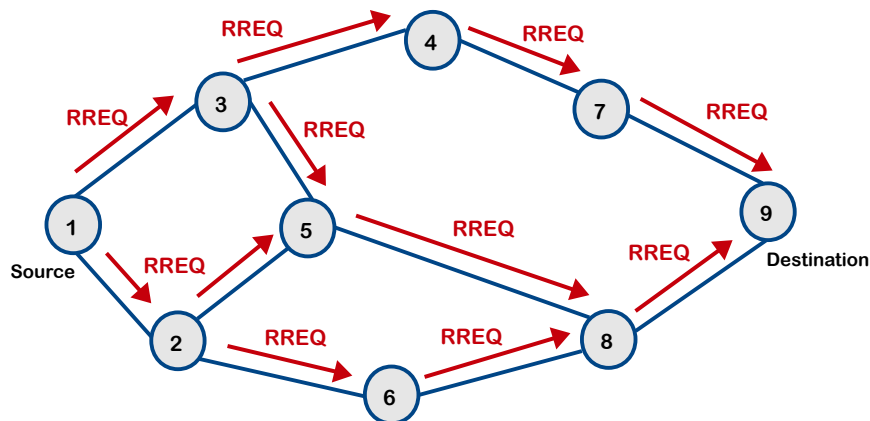


Figure 3.3. RREQ packets flooding AODV.

Each node maintains a cache to keep store RREQs that it has received and also stores the path back to each RREQ originator. When the destination or a node that has a path to the destination receives the RREQ, it checks the destination sequence numbers that it currently knows which is specified in the RREQ. To guarantee the freshness of routing information, *Route Reply (RREP)* packet (see Figure 3.4) is created and forwarded back to the source only if the destination sequence number is equal or greater than the one specified in RREQ. AODV uses only symmetric links and a RREP follows the reverse path of the respective RREQ. Upon receiving RREP packet, each intermediate node along the path updates its next hop table entries with respect to the destination node. Redundant RREP packets or with lower destination sequence number will be discarded.

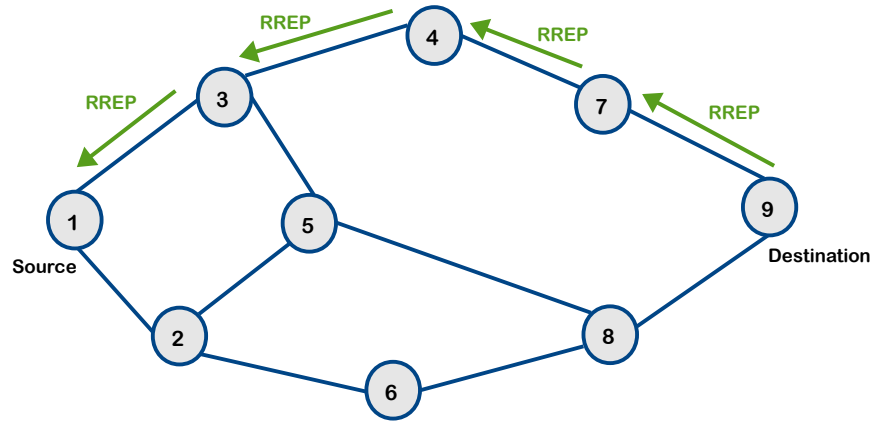


Figure 3.4. RREP packets in AODV.

3.3.1.2 AODV Route Maintenance

A node uses *hello messages* to notify its existence to its neighbours. Therefore, the link status to the next hop in an active path can be monitored. Whenever that a node discovers a link disconnection, it broadcasts a *Route Error (RERR)* packet to its neighbours, which in turn propagates RERR packet towards nodes whose paths may be affected by the disconnected link. Then, affected source can reinitiates a route discovery operation if the path is still needed.

3.4 Available Bandwidth Estimation (ABE)

In [Sarr 2008] an *Available Bandwidth Estimation* protocol (ABE) was proposed to estimate the available bandwidth in order to offer QoS, which is based on the basic operation of AODV regarding to discovery and maintain routes. AODV has the advantage of being simple and widely implemented.

That is, AODV-ABE is a modification of AODV routing protocol that includes ABE in its operation to estimate the available bandwidth on wireless links. When a source node requires to forward a packet to a destination, AODV-ABE floods a RREQ by including the required bandwidth in the RREQ. Each intermediate node that receives the RREQ checks if there is enough bandwidth. If there is enough bandwidth, RREQ is forwarded and it continues its course toward its destination. On the other hand, the required bandwidth can not be satisfied, the RREQ is discard. This process allows to establish the forwarding path that satisfies the required bandwidth.

3.4.1 ABE Operation

IEEE 802.11 [IEEE 802.11] is the standard adopted for VANETs in particular the extension IEEE 802.11p [IEEE 802.11p]. The only difference between them is that IEEE 802.11p provides specific extensions on the link layer regarding codification, frequency and so on. However, the MAC layer in IEEE 802.11p is exactly the same as in IEEE 802.11.

To improve the accuracy on AODV-ABE [Sarr 2008] the authors start from take into account the influence on the bandwidth available from a node to its neighbours:

- Carrier sense mechanism prevents two sources from transmitting simultaneously unless they have the same backoff time. The channel utilization has to be monitored to evaluate the capacity of a node to forward traffic.
- The available bandwidth on a link depends on both peers channel utilization ratios and on the idle periods synchronization, it also needs to be evaluated.
- Whenever a collision happens, both colliding packets are completely transmitted, maximizing the bandwidth lost.
- When a collision happens, it automatically retransmits the same packets drawing the backoff in a double sized *CW (Contention Window)*. The time lost in the additional overhead has an impact on the available bandwidth and also has to be evaluated.

3.4.1.1 Estimating Nodes Emission Capabilities

When a node wants to start a transmission, it needs to contend for medium access and it cannot transmit packets if the medium is busy. Therefore, a sender needs to evaluate the load of the medium. That is, the proportion of the time that the medium is idle to determine the opportunity that it has to successfully gain access to the shared resource. It is used the following notations:

- $T_{idle}(s)$ is the total idle time, it means the total time during which node s neither transmits any packet nor senses the medium busy.
- B_s is the bandwidth available to node s , it means the maximum throughput that it can use without disrupting already ongoing flow in the network.
- C_{max} is the medium capacity.
- Δ interval of seconds.

During an arbitrary observation interval Δ each node may monitor the sensing range and measure the total amount of time T_{idle} that is idle for transmit packets. They consider that this value is an upper bound of the available bandwidth [Sarr 2008]:

$$B_s \leq \frac{T_{idle}(s)}{\Delta} \cdot C_{max} \quad (3.1)$$

3.4.1.2 Estimating Link Available Bandwidth

It has been evaluated an upper bound of the available bandwidth a node could use to transmit packets. The reception side of the transmission also requires the medium to be free during the transmission, and thus, the previous measurement should also be considered at the receiver side. It has been introduced the following additional notations:

- δ is the time sampling step, it is the time unit.
- $\tau_m = \Delta / \delta$ is the number of time units in a measurement period.
- τ_s (respectively τ_r) is the number of time units during which the medium is available for node s (respectively r) in a measurement period, computed according to the constraints.
- B_s (respectively B_r) is the available bandwidth bound for node s (respectively r) measured with the formula (3.1) described in section 3.4.1.1.
- $B_{(s,r)}$ is the true available bandwidth on link (s, r) , it means the real bandwidth that can be achieved without degrading the other flows.
- $b_{(s,r)}$ is the estimated available bandwidth on link (s, r) .

When B_s is null or closer to zero, s does not have access to the medium. Similarly, if the medium is always busy on the receiver side, the packets experience collisions and the communication never succeeds. We can affirm that $B_{(s,r)} \leq \min(B_s, B_r)$. However, if sending a flow with a throughput higher $\min(B_s, B_r)$ it provokes a medium saturation around s and r . Considering this minimum value as the available bandwidth may also lead to an overestimation. In the general case, idle periods at transmitters and receivers sides are desynchronized. Due the complex interactions between nodes, transmitters and receivers need to be perfectly synchronized. Evaluating the impact of this asynchronism requires the exchange of the medium utilization patterns from both peers and a synchronization mechanism, which represents an extra overhead. They propose to use a probabilistic mechanism to estimate this effect.

Considering an uniform random distribution of the medium occupancy, the available expected bandwidth $E(b_{(s,r)})$ can be evaluated by expressing the probability that the medium is free simultaneously at the transmitters and receivers sides [Sarr 2008]:

$$E(b_{(s,r)}) = \sum_{i=0}^{\min(\tau_s, \tau_r)} i \cdot P(b_{(s,r)} = i) = \tau_s \times \tau_r \quad (3.2)$$

3.4.1.3 Estimating the Collisions Probability

The use of previous probabilistic estimation still leads to a certain level of inaccuracy. There is an opportunity for a packet that when is transmitted, the medium is not idle at the receiver side, provoking a collision. Transmitters can evaluate the collision probabilities toward certain receivers by counting the number of retransmissions events at the MAC layer. However, such strategy is only applicable to nodes already transmitting data. The evaluation mechanism should be active even when not data traffic is transmitted.

A collision probability may be computed on the bases of hello messages. When such messages are transmitted regularly, a receiver may estimate the amount of hello messages it should receive in a given time interval. Comparing this number with the effective number of received hello messages gives an estimation of the collision probability between both peers [Sarr 2008].

An imprecision comes from the size of these control messages. In a slowly evolving neighbourhood, successive hello messages are of comparable sizes. As a consequence, the computed collision probability may not be representative for small or big data packets. To address this issue, the authors in [Sarr 2008] extend the measurement by computing Lagrange interpolating polynomial appropriate the data. It designated by p_{hello} the collision probability computed on hello messages. And It is denoted by $f(m)$ this polynomial, the collision probability p_m for packets of m bits is approximated by [Sarr 2008]:

$$p_m = f(m) \cdot p_{hello} \quad (3.3)$$

3.4.1.4 Estimating the Exponential Backoff Time

When collisions happen, exponential backoff mechanisms is triggered. After each unsuccessful transmission, the contention window size is doubled up to a maximum value denoted by CW_{max} . In this situation, the average backoff value increases above $CW_{min} - 1$ divided by 2. And it is necessary to model the time consumed by the exponential backoff process.

For every packet, the transmission is successful at the first attempt with probability $(1 - p)$. It succeeds at the second attempt with probability $p \cdot (1 - p)$. After C unsuccessful retransmissions attempts, the IEEE 802.11 standard specifies that the packet should be dropped, resulting in at most $C + 1$ transmissions of the same packet.

It is denoted by X the random variable representing the number of retransmissions suffered by a given packet. And define M so that $CW_{min} = 2^M$. CW_{min} with $M \leq C$. It is obtaining the following formula:

$$\overline{backoff} = \frac{CW_{min} \cdot (1 - p) \cdot (1 - (2 \cdot p)^{M+1})}{2 \cdot (1 - 2 \cdot p)} + \frac{1}{2} \cdot (p^{M+1} - 1 + (CW_{max} - 1) \cdot (p^{M+1} - p^{C-M-1} + p^C)) \quad (3.4)$$

K is denoted by the proportion of bandwidth consumed by the backoff mechanism when collisions happen and by $T(m)$ the time separating the emission of two consecutive packets. This delay depends on the emission rate on the frame size m and DIFS [IEEE 802.11] is a fixed interval. Then, K can be expressed by the following notation [Sarr 2008]:

$$K = \frac{DIFS + \overline{backoff}}{T(m)} \quad (3.5)$$

3.4.2 ABE Calculation

Finally, the available bandwidth estimation between two neighbour nodes s and r can be estimated by the following equation [Sarr 2008]:

$$E_{final}(b_{(s,r)}) = (1 - K) \cdot (1 - p) \cdot E(b_{(s,r)}) \quad (3.6)$$

Where $E(b_{(s,r)})$ is the available bandwidth on link (s, r) evaluated by monitoring the radio channel and combining transmitter and receiver values in a probabilistic manner, p is the collision probability measured on the received hello messages and K is the proportion of bandwidth lost due to the backoff scheme computed due to p . For further information regarding to ABE it can refer to [Sarr 2008].

The different issues aforementioned can be combined to estimate the available bandwidth in a wireless link. That is between a given transmitter and receiver. All this whole mechanisms describes lead AODV-ABE to be a lightweight protocol as it mainly relies on the perception that nodes have of their immediate environment.

For this project, we have implemented in NCTUns [NCTUns] and extension of AODV [Perkins 1999] that includes ABE. In [Tripp] there is a preliminary performance evaluation of AODV-ABE that show simulation results regarding the throughput obtained with an scenario designed for static nodes and another scenario for mobile nodes. The main goal of this project is to carry out a performance evaluation of AODV-ABE in VANETs environments specifically in urban scenarios.

The following chapter describes the capabilities and features, components and architecture, simulation environment and simulation settings of NCTUns simulator [NCTUns] and Citymob mobility pattern generator [Citymob 2008] which are the tools selected in this project to carry out a performance evaluation to compare AODV and AODV-ABE.

4. SIMULATION ENVIRONMENT NCTUns AND CITYMOB

4.1 Introduction to NCTUns

NCTUns (National Chiao Tung University network simulator) is a network simulator capable of simulating various wired and wireless IP networks. In addition to being a network simulator, NCTUns is an emulator useful for testing real-life network devices. The core technology of NCTUns is based on a simulation methodology invented by Prof. S.Y. Wang [NCTUns 2010].

NCTUns is a tool for researchers to evaluate the performances of real-life applications. It directly uses the real-life network protocol stacks on Linux operating system for realistic network simulations and enables any real-life network application to be executed on a node in a simulated network. NCTUns is an open-source software running on Linux Fedora 12 with an integrated GUI environment.

4.1.1 Capabilities and Features

NCTUns is composed by a kernel-reentering simulation methodology, which it provides several unique advantages that cannot be easily achieved by traditional network simulators. The following are described its capabilities and features [NCTUns]:

- **High-Fidelity Simulation Results.** NCTUns uses the real-life Linux TCP/IP protocol stack to generate high-fidelity simulation results. By using the kernel re-entering simulation methodology, a real-life UNIX kernel protocol stack is directly used to generate high-fidelity simulation results.
- **Reusing All Real-Life Application Programs.** Real-life existing UNIX applications programs can be run up on a node in a simulated network. This provide several advantages:
 - Generate realistic network traffic to drive simulations, which leads to more convincing results than using artificial traffic generated by simple functions.
 - The performances of these real-life applications under various network conditions can be evaluated and then improved before they are released to the public.
 - The applications developed at the simulation study stage can be readily used and deployed on real-life UNIX machines when the simulation study is finished.
- **Same Configuration and Operation as for Real-Life Networks.** The configuration and operation for a simulated network are exactly the same as those for real-life IP network. This provides two advantages:
 - User knows how to configure and operate a real-life IP network, immediately knows how to configure and operate a simulated network in NCTUns.
 - Conversely, NCTUns can be used as a training tool to learn how to configure and operate a real-life IP network.
- **Seamless Integration of Emulation and Simulation.** NCTUns can be turned into an emulator easily. In an emulation, nodes in a simulated network can exchange real packets with real-world machines via the simulated network. This capability is very useful for testing the functions and performances of real-life device. For instance, VoIP phone, fixed host, mobile host, router and so on.
- **High Simulation Speeds and Repeatable Simulation Results.** NCTUns combines the kernel re-entering simulation methodology with the discrete event simulation methodology. It modifies the process scheduler of the Linux kernel to accurately control the execution order of the simulation engine process and all involved real-life application processes. If the same random number seed is used for a simulation case, the simulation results are repeatable across different runs.
- **Support for Various Important Networks.** NCTUns supports the following networks environments:
 - Ethernet-based IP networks with fixed nodes and point-to-point links.
 - IEEE 802.11 (a) and (b) wireless LAN networks, including both the ad hoc and infrastructure.

- GPRS cellular networks.
- Optical networks, including traditional circuit switching network and more advanced *Optical Burst Switching (OBS)* networks.
- IEEE 802.11 (b) wireless mesh networks.
- IEEE 802.11 (e) QoS networks, tactical and active mobile ad hoc networks, and wireless networks, with directional and steerable antennas.
- IEEE 802.16 (d) WiMAX networks, including the PMP and mesh modes.
- IEEE 802.16 (e) mobile WiMAX PMP networks.
- IEEE 802.16 (j) transparent mode and non-transparent mode relay WiMAX networks.
- DVB-RCS satellite networks for a GEO satellite located 36 000 km above the earth.
- **Support for Various Networks Devices.** NCTUns simulates common networking devices such as Ethernet hubs, switches, routers, hosts, IEEE 802.11 (a) and (b) wireless access points and interfaces.
 - For optical networks, it simulates optical circuit switches and optical burst switches, WDM optical fibers, and WDM protection rings.
 - For DiffServ QoS networks, it simulates DiffServ boundary and interior routers for QoS provision.
 - For GPRS networks, it simulates GPRS phones, GPRS base stations, SGSN and GGSN devices.
 - For 802.16(d) WiMAX networks, it simulates the PMP-mode base stations (BS) and Subscriber Stations (SS) and the mesh-mode base stations and Subscriber Stations (SS).
 - For 802.16(e) WiMAX networks, it simulates the PMP-mode base stations (BS) and Subscriber Stations (SS).
 - For 802.16 (j) transparent mode and non-transparent mode WiMAX networks, it simulates the base stations (BS), relay stations (RS), and mobile stations (MS).
 - For DVB-RCS network, it simulates the GEO satellite, Network Control Center (NCC), Return Channel Satellite Terminal (RCST), feeder, service provider, traffic gateway.
 - For wireless vehicular networks,
 - ITS cars are equipped with an 802.11 (b) ad hoc mode and infrastructure mode wireless interface.
 - ITS cars are equipped with GPRS wireless interface.
 - ITS cars are equipped with a DVB-RCST wireless interface.
 - For mobile nodes are equipped with multiple heterogeneous wireless interfaces it simulates:
 - A traditional mobile node that moves on a pre-specified path (e.g., random waypoints).

- An ITS car that automatically move (auto-pilot) on a constructed road.
- An import mobile nodes and their paths from file, created by an external mobility pattern generator (.mdt).
- **Support for Various Network Protocols.** NCTUns simulates various protocols such as:
 - IEEE 802.3 CSMA/CD MAC.
 - IEEE 802.11 (a)(b)(e) CSMA/CA MAC.
 - Learning bridge and spanning tree protocols used by switches, IP, Mobile IP, RIP, OSPF, UDP, TCP, HTTP, FTP, Telnet, AODV, DSR, etc.
 - DiffServ QoS protocol suite, the optical light-path setup protocol, the RTP/RTCP/SDP protocol suite.
 - IEEE 802.16 (d)(e)(j) WiMAX PMP protocol suites and the IEEE 802.16 (d) mesh mode protocol suite.
 - DVB-RCST protocol suite.
- **Highly-Integrated and Professional GUI Environment.** NCTUns GUI program has the capacity of:
 - Drawing network topologies.
 - Configuring the protocol modules used inside a node.
 - Configuring the parameter values used inside a protocol module.
 - Specifying the initial locations and moving paths of mobile nodes.
 - Plotting network performance graphs.
 - Playing back the animation of a logged packet transfer trace.
 - Pasting a map graph on the background of the network topology.
 - Constructing a road network for wireless vehicular network simulations.
- **Popular Operating System Support.** NCTUns runs on Linux operating systems. The Linux distribution that NCTUns 6.0 currently supports is Red-Hat Fedora 12, whose Linux Kernel version currently is 2.6.31.6.
- **Open-System Architecture.** By using a set of well-defined module APIs that are provided by the simulation engine, protocol module developed can easily implement own protocol and integrate it onto the simulation engine.
- **Distributed Architecture for Remote and Concurrent Simulations.** By using a distributed architecture, each component of NCTUns can be run on separate machine. As such, the machine that runs the GUI program can be different from the machine that runs the simulation engine. This capability sometimes can have an advantage; when simulating a very large case with hundreds of mobile nodes, the GUI will consume many CPU cycles to draw the movements of these mobile nodes during the simulation.

Although IEEE 802.11p [IEEE 802.11p] is the particular standard extension adopted for VANETs that includes specific features for the physical layer (e.g., codification, frequency) to cope the high mobility issue. However, the MAC layer in IEEE 802.11p is exactly the same and has the same basic operation as in IEEE 802.11b [IEEE 802.11]; the only difference is that IEEE 802.11p improves the physical layer for high speeds, so it could expect even better results with IEEE 802.11p. Nonetheless, IEEE 802.11p is not yet implemented in NCTUns 6.0.

The commercial version of NCTUns is called EstiNet Network Simulator and Emulator, is *“a software tool for network planning, testing, education, protocol development and applications performance prediction. It is both a network simulator and emulator with worldwide customers and global impact”* [EstiNet 2010].

4.1.2 Architecture and Components

NCTUns is a distributed architecture which it is conformed by eight principal components [NCTUns] which are explained below in this section. In Figure 4.1 shows a generic NCTUns architecture that describe the main functions of the components.

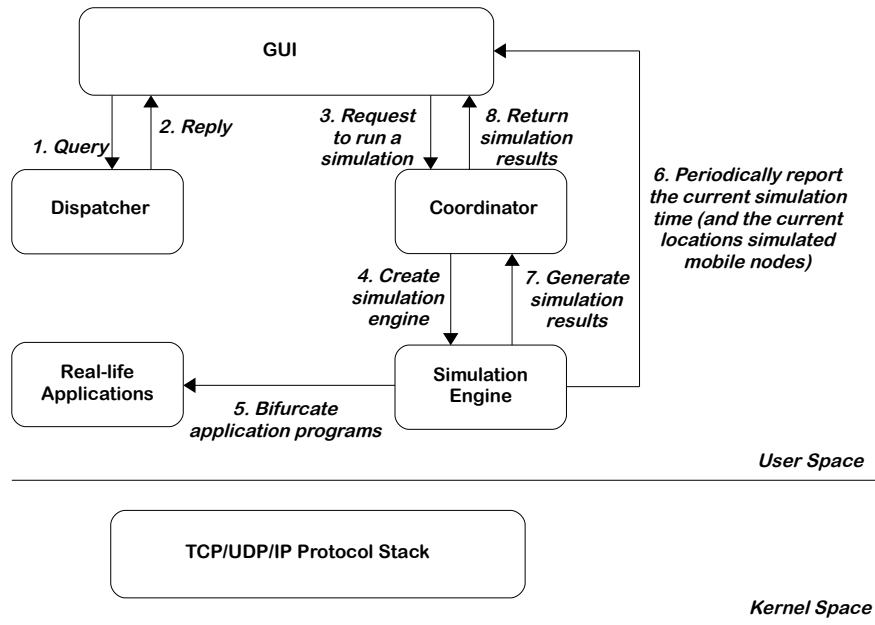


Figure 4.1. NCTUns generic architecture.

- 1) *GUI program* by which it can edit a network topology, configures the protocol modules used inside a network node, specifies mobile nodes initial locations and moving paths, plots performance graphs, plays back the animation of a packet transfer trace, and so on.
- 2) *Simulation engine program* which provides basic and useful simulation services (e.g., event scheduling, timer management, packet manipulation, etc.) to protocol modules.
- 3) Set of various *protocol modules*, each of which implements a specific protocol or function (e.g., packet scheduling or buffer management). All protocol modules are C++ classes and are compiled and linked with the simulation engine program.
- 4) Simulation job *dispatcher program* that can simultaneously manage and use multiple simulation servers to increase the aggregate simulation performance. It can be run on a separate machine or on a simulation server. In other words, dispatcher program is responsible for monitoring the statuses of the simulation servers that it manages and selecting an available simulation server to serve the simulation request issued from the GUI program.
- 5) *Coordinator program* on each simulation server it must be run up. Coordinator should be work as long as the simulation server is working. This enables the dispatcher to choose a simulation server from its available servers to service a job. When the coordinator receives a job from the dispatcher, it bifurcates a simulation engine process to simulate the specified network and protocols. It may also bifurcate several real-life application program processes specified in the job. These processes are used to generate traffic in the simulated network.
- 6) *Kernel patches* that need to be made to the kernel source code so that a simulation engine process can run on a UNIX machine correctly. Currently NCTUns 6.0 run on Red-Hat Fedora 12, which uses the Linux 2.6.31.6 kernel.
- 7) Various *real-life user-level application programs*. Due the kernel-reentering simulation methodology, any real-life existing or to be developed application program can be directly run up on a simulated network to generate realistic network traffic.
- 8) Various *user-level daemons* that are run up for the whole simulation case. For example, NCTUns provides RIP and OSPF routing daemons. By running these daemons, the routing entries needed for a simulated network can be constructed automatically.

Due to this distributed design, a remote user can submit its simulation job to a job dispatcher, and the dispatcher forwards the job to an available simulation server for its execution. The server will simulate the job and later return the results back to the remote GUI program for further analysis. This scheme can easily support multiple simulation jobs performed concurrently on different simulation servers. Figure 4.2 shows the distributed architecture of NCTUns where *Simulation Service Center* is composed by: kernel modifications, simulation engine, protocol modules and coordinator.

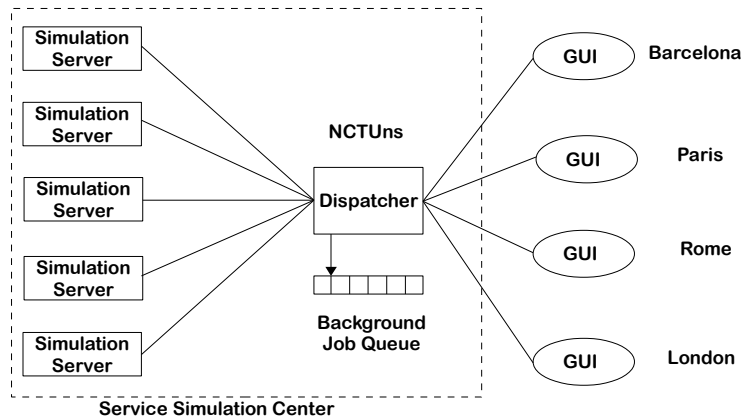


Figure 4.2. NCTUns distributed architecture.

4.2 Installation and Execution of NCTUns

Each release of NCTUns is developed for specific version of Fedora Core. The reason of this requirement is the modifications that simulator installer does on certain parts of Linux kernel. As it is aforementioned in section 4.1.2, kernel patches need to be made in the kernel source code so that a simulation engine process can run on a UNIX machine correctly.

In the development of this project, we installed NCTUns on a virtual machine by using VMware Fusion for Mac OS X, according the current version NCTUns 6.0 over Red-Hat Fedora 12 implemented on linux kernel 2.6.31.6.

The computer used to perform the simulations has the following characteristics:

- Operating System: Mac OS X (Leopard) version 10.5.8
- Processor: 2.4 GHz Intel Core 2 Duo
- Memory: 2 GB 667 MHz DDR2 SDRAM

4.2.1 Installation Steps

At first place, we have to download VMware Fusion available on [VMware 2011] and install it according the available guide steps on [VMware 2009]. At second place, we have to download Fedora 12 available on [Fedora 2009a] and install it according the available guide steps on [Fedora 2009b]. Finally, NCTUns 6.0 for Fedora 12 can be downloaded at the following url: <http://nsl.csie.nctu.edu.tw>

Once installed VMware Fusion and Fedora 12. Before installing NCTUns some dependences must be installed, logged in a terminal as root in order to install the packages, we have to type the following commands:

```
yum groupinstall "Development Libraries" "Development Tools" "Legacy Software Development"
yum install mkinitrd -y xterm
```

After install these dependences, we can decompress the NCTUns distribution and execute the `install.sh` script. The software is installed in the directory: `/usr/local/nctuns`. And the tunnel interfaces are created in the `/dev` directory.

During the installation process, the installer asks if a `nctuns` user must be created, if the kernel has to be patched, if *SELinux* (*Security Enhanced Linux*) should deactivated. For all the questions we have to answered yes.

Once that this process is finished, the system must be rebooted and started with the parched kernel by selecting NCTUns (2.6.31.6-nctuns20091227) in the GRUB bootloader. Which is the option what we will always use as shown in Figure 4.3.

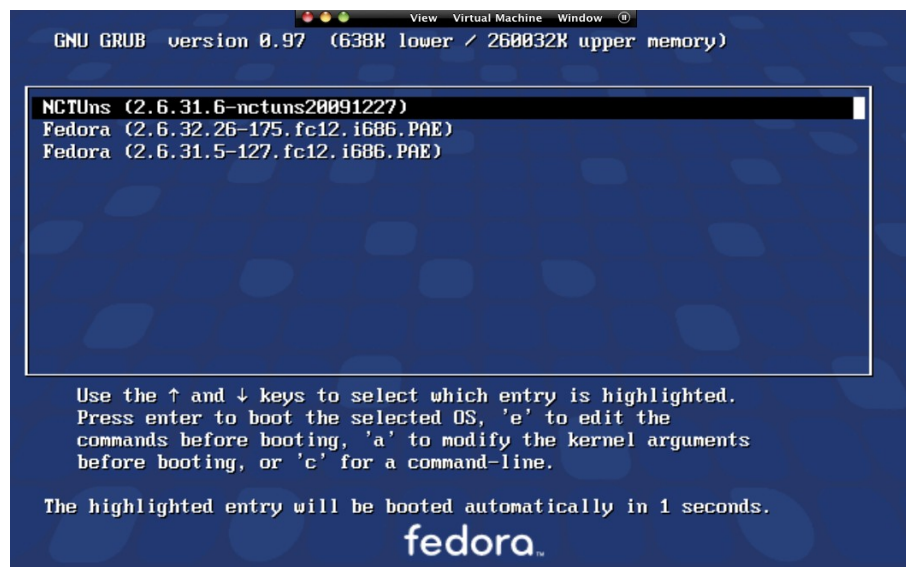


Figure 4.3. GRUB bootloader.

Then we can login with the following username and password, which is the login what we will always use, as shown in Figure 4.4.

```
username: nctuns  
password: nctuns
```



Figure 4.4. Login screen nctuns user on fedora.

The last step to complete the installation of NCTUns is configure the shell preferences that include the NCTUns environment variables in order to quick access to application. By type the following lines in `.bashrc` file in the directory: `/home/nctuns/bashrc` by adding the following commands:

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

After the execution of these steps, we have to reboot the system and we already able to execute and start working in NCTUns simulator.

4.2.2 Starting NCTUns

NCTUns provides a front-end GUI program (called “nctunsclient” in its package). NCTUns need two more components in order to management the simulation resources has it is described in section 4.1.2 regarding to dispatcher and coordinator. These three components have to be executed in a terminal as root. The best way to execute these components is to execute it in different terminals since, each of them echoes information about the simulation process that can be useful to debug if there is a problem.

At this way in different terminals and logged as root, we have to execute the commands in the same order as follows:

```
dispatcher &
coordinator &
nctunsclient &
```

Once the dispatcher, coordinator and nctunsclient have been launched the following starting screen as shown in Figure 4.5 will pop up and NCTUns is now ready to start working.



Figure 4.5. Screenshot NCTUns.

4.2.2.1 Operation Modes

In the GUI, there are four operation modes D, E, R and P as shown in Figure 4.6. Which are described below:



Figure 4.6. Operation modes

- 1) *“Draw Topology” mode.* In this mode it can insert network nodes, create network links, and specify the locations and moving paths of mobile nodes. In addition, the GUI program provides a complete tool kit for user to construct road network, which is fundamental to wireless vehicular network simulations.
- 2) *“Edit Property” mode.* In this mode it can double click the icon of a network node to configure its properties (e.g., the network protocol stack used in this node, the applications to be run on this node during simulation, and other parameters).
- 3) *“Run Simulation” mode.* In this mode, the GUI program provides user with a complete set of commands to start/pause/stop/abort the simulation. It can easily control the progress of a simulation by pressing a button on the GUI control panel.
- 4) *“Play Back” mode.* After a simulation is finished, the GUI program will automatically switch itself into the “Play Back” mode and read the packet trace file generated during the simulation. In this mode, it can use the GUI program to replay the packet transmission/reception operation in an animated way.

For further information about NCTUns operation modes and tools a GUI user manual for NCTUns 6.0 is available on [NCTUns].

4.3 Design VANET urban scenario

This section described the design and dimensions of the urban scenario implemented to this project, which is an scenario based on Manhattan Downtown model. Similar to Eixample district of Barcelona.

In the first place, being in “Draw Topology ” mode. It selects ITS road segment and ITS crossroad as shown in Figure 4.7 and 4.8 in order to draw the streets of the scenario. It select 2 lanes on a road, with one lane per each direction and a lane width is 20 meters. Therefore, the road with 2 lanes is 40 meters per street.



Figure 4.8. ITS road segment.



Figure 4.7. ITS crossroad.

In Figure 4.9 Obstacles simulate the buildings in the streets which may block wireless signal, view and/or movement. The obstacles are configured to suffer an attenuation from 5 dB according a survey of propagation models in VANETS in [Eenennaam 2008] caused by buildings according the environment urban area with *NLOS (Non Light Of Sight)*.

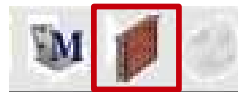


Figure 4.9. Obstacles

As shown in Figure 4.10, there is a screenshot scenario with the streets, intersections and obstacles as buildings. However, the vehicles allocation on the streets are designed using an external movement pattern generator, Citymob [Citymob 2008] explained later on section 4.4.

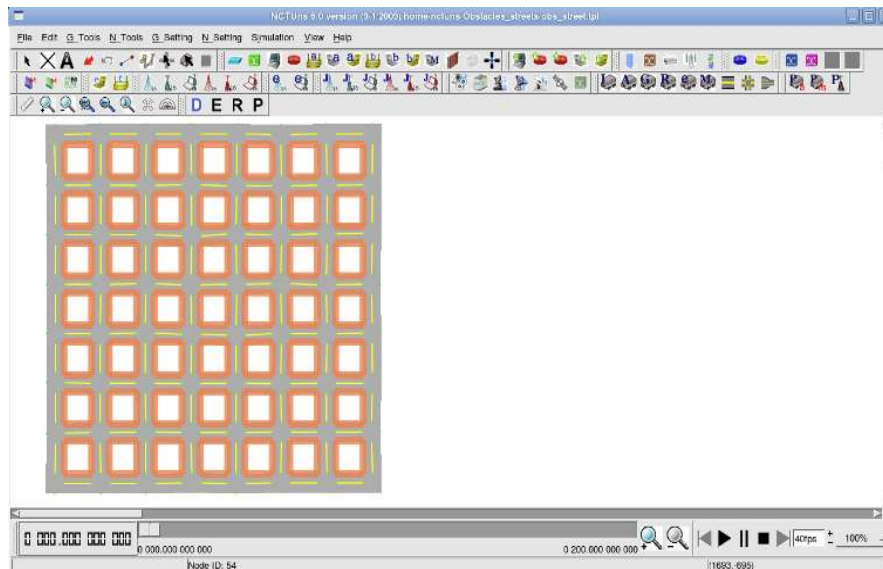


Figure 4.10. Screenshot scenario: streets, intersections and buildings.

The scenario dimensions are depicted and described as shown in Figure 4.11. There is a scenario of 1000 x 1000 meters with a distance between streets of 100 meters and with a width per each lane of 20 meters therefore, there are 2 lanes one per each direction and the road with 2 lanes is 40 meters per street.

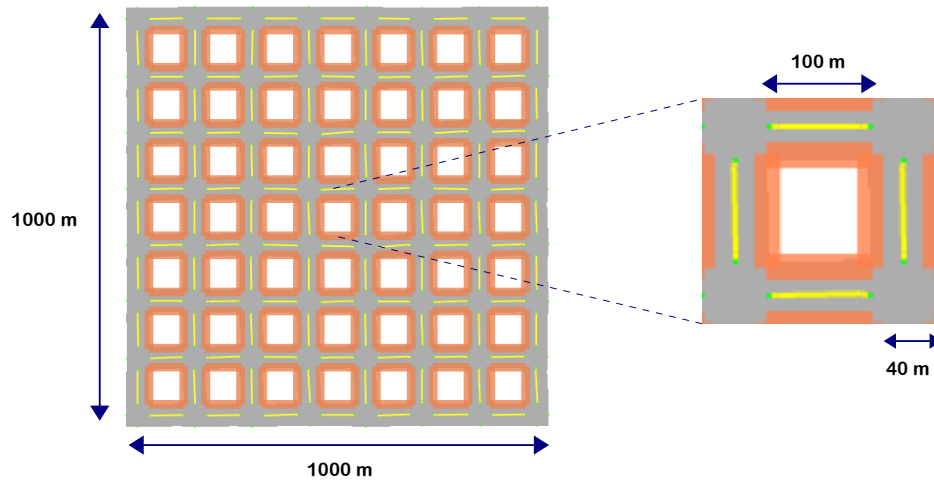


Figure 4.11. Scenario dimensions.

This is the urban scenario designed what we implemented to perform the simulations. However, as it is mentioned on the introduction of the project the mobility pattern generator is carried out by Citymob. On next section it is introduced Citymob after that, it is returned to this scenario in order to established the simulation settings.

4.4 Introduction to Citymob

Citymob is a mobility pattern generator developed in the Universidad Politécnica de Valencia [Citymob 2008]. Especially designed to investigate different mobility model in VANETs and their impact on inter-vehicle communication performance and it creates urban mobility scenarios.

4.4.1 Mobility Models

Citymob proposes three different mobility models that combine a certain level of randomness, while trying to represent some realistic environments. The models are [Citymob 2008]: *Simple Model (SM)*, *Manhattan Model (MM)*, *Downtown Model (DM)*.

- *Simple Model (SM)*. Models vertical and horizontal mobility patterns without direction changes. Semaphores are no supported either.
- *Manhattan Model (MM)*. Model the city as a Manhattan style grid, with a uniform block size across the simulation area. All streets are two-way, with one lane in each direction. Vehicles movements are constrained by these lanes. The direction of each vehicle in every moment will

be random, and it can not be repeated in two consecutive movements. Moreover, this model simulates semaphores at random positions (not only in crossing), and with different delays. When a vehicle finds a semaphore, it will remain stopped until the semaphore turns green.

- **Downtown Model (DM).** This model adds traffic density to the Manhattan Model. In a real town, traffic is not uniformly distributed; there are zones with a higher vehicle density. These zones are usually in the downtown, and vehicles must move more slowly than in the outskirts. The downtown area is defined by the coordinates (*start_x*, *end_x*, *start_y*, *end_y*) and can never cover more than 90% of the total map area. Parameter *p* is used to establish the probability of a node being initially located inside the downtown area, and also the probability that nodes on the outskirts move into the downtown. Once a vehicle enters this area, it will move slower. The remaining characteristics are the same as for the Manhattan Model.

4.4.2 Citymob Guidelines

Simulation framework generates mobility traces for these three mobility models proposed according to the following guidelines [Citymob 2008]:

- The city is simulated in the same way for the three models, with a Manhattan grid map. Map width and height are configurable parameters.
- The distance between streets is also configurable, logically limited by map size. There must be a minimum number of crossings to allow vehicles to change their direction.
- Every vehicle will start at a random position inside the map, although in the Downtown model the probability of starting inside the downtown is greater.
- Speed can vary according to map area, changing throughout the simulation. Every vehicle will travel with a random speed for each movement, always lower than the maximum speed defined.
- The presence of semaphores (for the Manhattan and Downtown Model). Vehicles will simulate semaphores by stopping themselves randomly. This way, simulation is more realistic since in a city semaphores are not systematically distributed on streets, and it also helps to model other unforeseen traffic events for instance vehicle suddenly stops.

4.4.3 Installation and Operation

Citymob simulator can be downloaded from the website of the *Grupo de Redes de Computadoras (GRC)* in [Citymob 2009]. Citymob has been implemented using the C programming language and it is distributed under a GNU/GPL license.

The use of Citymob is very simple. Only have to execute the application followed by the required parameters in order to generate the desired trace file.

In a terminal, in the directory where Citymob is placed, execute the following command:

```
./citymob -m M -n N -t T -s S -w W -h H -d D -a A -x x1 -y y1 -X X1 -Y Y1 -p P
```

Where,

M: Model

M=1: Simple Simulation Traffic Model.

M=2: Manhattan Simulation Traffic Model.

M=3: Downtown Simulation Traffic Model.

N: Vehicles number.

T: Simulation time.

S: Max speed.

W: Map width.

H: Map height.

D: Streets distance.

A: Accidents number.

x1: min x downtown (only for M=3).

y1: min y downtown (only for M=3).

X1: max X downtown (only for M=3).

Y1: max Y downtown (only for M=3).

P: Probability of a node being initially located inside the downtown (only for M=3).

4.5 *Design VANET urban movement*

The scenarios chosen for the simulations of this project has the following characteristics:

- **Manhattan Downtown Model.**
- **20, 40, 60, 80, 100 vehicles.**
- **Simulation time 200 seconds.**
- **Speed 14m/s (50km/h)**
- **Dimensions 1000 x 1000 meters.**
- **Distance between streets of 100 meters.**

- Downtown: 900 x 900 (according the condition that can not cover more than 90% of the total map area).
- Probability of a vehicle being downtown is set 1. That is all vehicles are into downtown initially.

At this way, we generate 5 different mobility pattern in order to obtain the corresponding mobility pattern for the 20, 40, 60, 80 and 100 vehicles. Which are the 5 different scenarios evaluating in this project. In a terminal we executed the following commands according the parameters explained in section 4.4.3.

```
./citymob -m 3 -n 20 -t 200 -s 14 -w 1000 -h 1000 -d 100 -x 100 -y 100 -X 900 -Y 900 -p 1
```

In order to generate the movement pattern we have to use the command indicated, unless otherwise indicated, the results are displayed on screen; as we are interested in storing the movement patterns in a file we must type the following command in a terminal:

```
./citymob [PARAMETERS] > prueba.txt
```

This is because, Citymob mobility pattern generator is not compatible with NCTUns simulator. Nevertheless, in [Campos 2011] which is a PFC developed by two students of the working group. There is, a translator program developed in order to be compatible and be able to import the vehicles and movement to NCTUns simulator. For this reason, the file must be name `prueba.txt` in order to be compatible with the translator program, so and then we have to type the following commands in order to export the results in the file named `prueba.txt`.

```
./citymob -m 3 -n 20 -t 200 -s 14 -w 1000 -h 1000 -d 100 -x 100 -y 100 -X 900 -Y 900 -p 1 > prueba.txt
```

This command was executed also with: `- n 40`, `- n 60`, `- n 80`, `- n 100`. And the rest of parameters are exactly the same in order to obtain the trace file for the 5 scenarios with different number of vehicles.

Once generated the file, it should be copied to the folder where the translator is located and we have to executed the following command:

```
./traductor
```

It will generate a file called `salida.mdt` that contains the vehicles and movement which is able to import to NCTUns simulator. In this case, we have obtained 5 different `salida.mdt` in order the 5 different scenarios according to 20, 40, 60, 80 and 100 vehicles.

For further information about the translator there is available in [Campos 2011].

4.6 Configuration of Simulation Environment Scenarios

Once we have designed VANET urban scenario according the Figure 4.10 in section 4.3. And we have the files `salida.mdt` according the section 4.5. We are able to import the files into NCTUns simulator. Being in “Drawn Topology” mode, as shown in Figure 4.12 from the menu “G_Tools” > “Import Mobile Nodes and Their Paths from File” we select the file `salida.mdt` obtained with the translator.

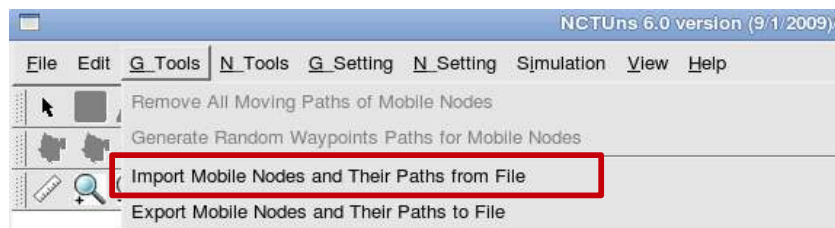


Figure 4.12. Import vehicles and movement

Is important to highlight that vehicles imported from Citymob are: 802.11(b) mobile node (ad hoc mode). As aforementioned in section 4.1.1, although 802.11p is the particular standard extension adopted for VANETs that includes specific features for the physical layer (e.g., codification, frequency). However, the MAC layer in IEEE 802.11p has the same basic operation as in IEEE 802.11b; the only difference is that IEEE 802.11p improves the physical layer, so it could expect even better results with IEEE 802.11p. Nonetheless, IEEE 802.11p is not yet implemented in NCTUns 6.0.

And then, after this step, we already have loaded into the NCTUns simulator the vehicles and its movement pattern generated by Citymob. As shown in Figure 4.13 according the scenario with 100 vehicles.

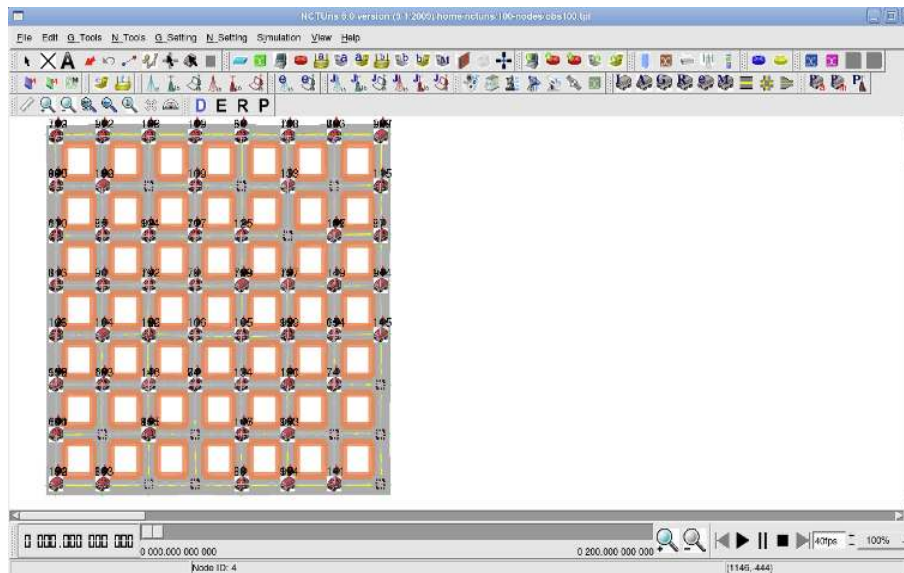


Figure 4.13. Screenshot scenario with 100 vehicles.

Now we proceed as described in section 4.2.2.1 to “Edit property” mode in order to configure the vehicles and its properties and parameters in order to perform the simulations. That is, the network protocol stack used, the applications to be run during the simulation and other parameters.



Figure 4.14. Edit property mode.

First of all, we select by double click any vehicle a menu called “mobile station” is open. Path tab allows adding movement to vehicle but, in our case, the mobility model is generated by the Citymob mobility pattern generator. In the same tab we can modify the protocol stack of the network interface by clicking on the “Node Editor” button. We have to delete, using the X button the GOD block and replace it with AODV block, which is located in the MROUTED tab, and reconnect the blocks with arrow button. The protocol stack must stay as below in Figure 4.15.

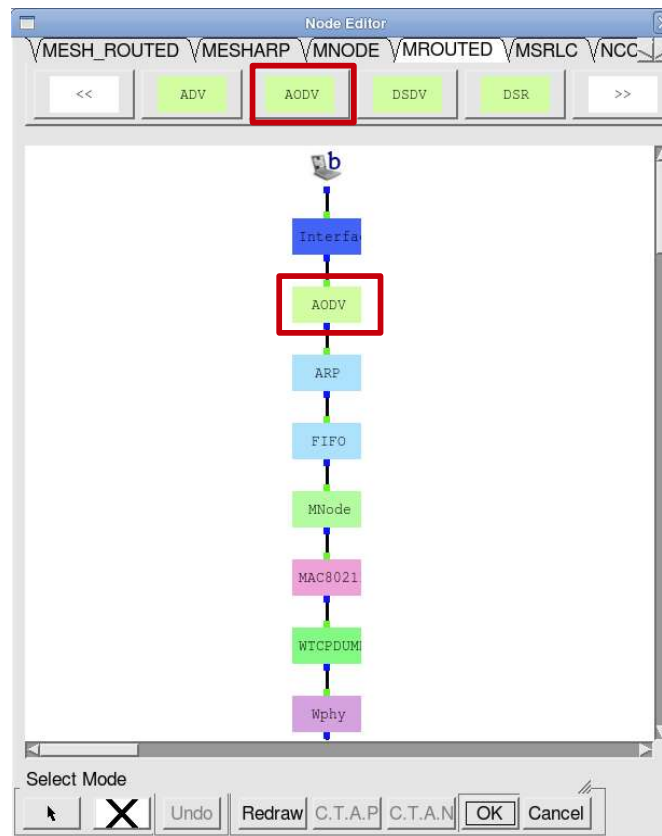


Figure 4.15. Protocol stack in the Node Editor.

Is important to highlight at this point that in order to correctly perform the simulations due the objective is to compare AODV and ABE routing protocol. Every time that we switch to perform simulations with AODV or ABE. And as it is mentioning at the beginning of section 3.4 that ABE is a modification of AODV routing protocol that includes ABE in its operation to estimate the available bandwidth on wireless links. We have to replace the files that contain ABE by AODV in order to switch them.

The original files to AODV routing protocol are the following four files: mac-802-dcf.cc, mac-802-dcf.h, AODV.cc and AODV.h. In order to include ABE routing protocol, we have the same files but modified including the algorithms of ABE operation.

That is, in a terminal logged as root in the corresponding path we have to replace the original AODV files by the AODV files modified which include ABE:

```
NCTUns-6.0/src/nctuns/module/mac/  
    mac-802-dcf.cc  
    mac-802-dcf.h  
  
NCTUns-6.0/src/nctuns/module/route/aodv  
    AODV.cc  
    AODV.h
```

After replace the files, we have to compile the files by executing the command `make all` and `make install` in the directory: `NCTUns-6.0/src/nctuns` in order to applied the changes in the files and perform the corresponding simulations with AODV routing protocol and ABE routing protocol. By configuring the corresponding protocol in the protocol stack in the node editor, as it is show in Figure 4.15.

After configuring the protocol stack, we have to click OK and copy this protocol stack to all vehicles by push the *C.P.A.N.S.T* (*Copy the nodes Protocol stack to All Nodes of the Same Type*) button:

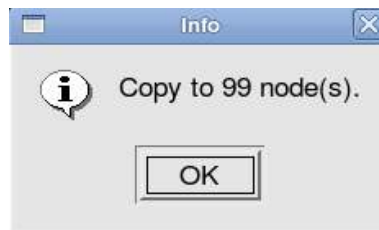


Figure 4.16. C.P.A.N.S.T. button.

In the “Application” tab we can add and configure the vehicles that will generate traffic and the vehicles that will receive the traffic. In the simulations of this project the network traffic is generated by command `stg` application program and the receiver uses command `rtg` application program.

For instance, it selects the vehicle 100, which it is generating traffic by command `stg -i` that indicates that traffic is import from file `TRACE1.config` and it is sending to vehicle IP address corresponding to vehicle 51 as shown in Figure 4.17. And on the Input file name it selects the Browse button and choose the path from the file.

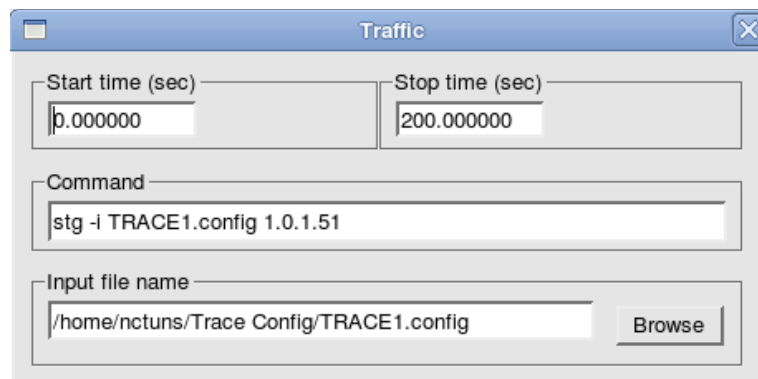


Figure 4.17. Command stg application program.

TRACE1.config file contains the following:

```
type: udp
start_time: 0
on-off: 1
on: time: 200 const 0.01 length: const 1000
end
```

It means that, during 200 seconds it is sending traffic type UDP in packets of 1000 bytes in a constant way separated each one of them by 0.01 seconds.

After configure the command stg application program and select the Input file name we still remain in the “Application” tab where we can add, modify or delete another command application programs. As shown in Figure 4.18. NCTUns features different built-in applications, which use can be consulted by clicking in the “App. Usage” button which provides command usage information for each pre-installed application program. A program usage information window will pop up shadowing the detailed usage for each pre-installed application program. For further information regarding to “App. Usage” it can refer to [NCTUns 2010].

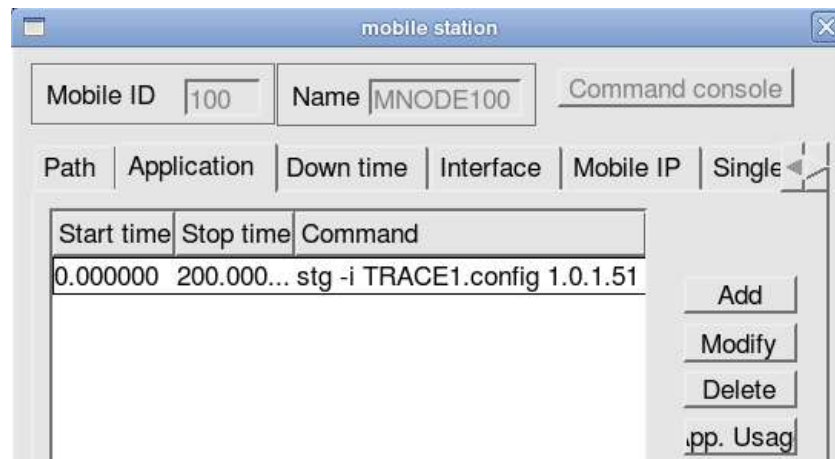
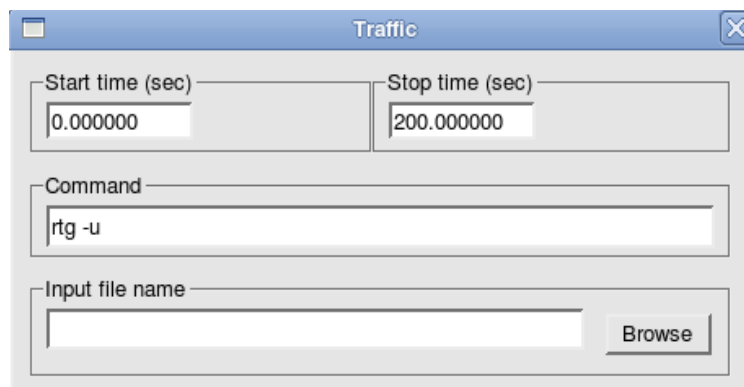


Figure 4.18. Add, modify and delete command application program.

By other the vehicle 51 which is received the traffic generated by the vehicle 100 is configured as by use the following command: `rtg -u`. As shown in Figure 4.19. This means, that it is receiving traffic UDP.

Figure 4.19. Command `rtg` application program.

At the same way after configure the command `rtg` application program we still remain in the “Application” tab where we can add, modify or delete another command application programs. As shown in Figure 4.20. NCTUns features different built-in applications, which use can be consulted by clicking in the “App. Usage” button.

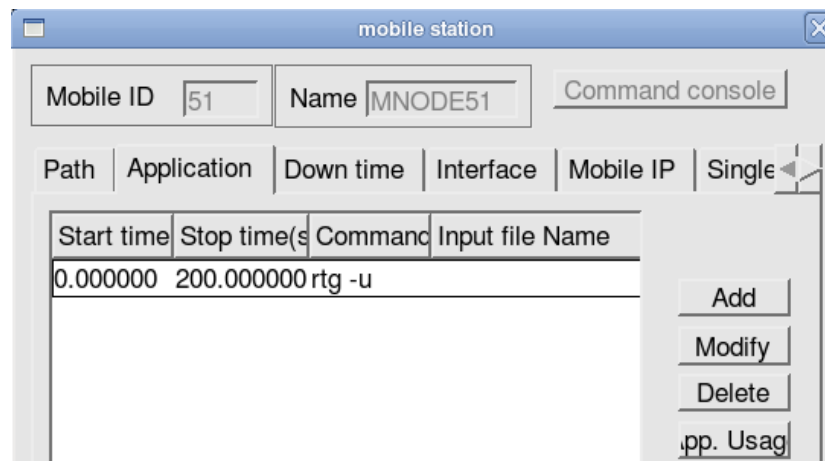


Figure 4.20. Add, modify and delete command application program.

For setting parameters related to the physical layer we can use the physical layer button. As shown in Figure 4.21.



Figure 4.21. Physical layer button.

By clicking in physical layer button the “Specify physical layer and channel model parameter” screen appears in order to configure the following parameters, as shown in Figure 4.22. We can configure “Propagation Channel Model”, “Node Connectivity Display” and “Node Connectivity Display” box.

It is selecting “Theoretical Channel Model” > Path Loss Model > “1: Two Ray Ground”. Because, Two Ray Ground reflection model considers both the direct path and a ground reflection path, at this way this model gives more accurate prediction at a long distance than the free space model. Fading Model > “1: Rayleigh”. Because, Rayleigh fading is most applicable when there is not dominant propagation along a line of sight between the transmitter and receiver.

“Node Connectivity Display” box it is selecting “Use the receiving node perspective”. And “Node Connectivity Determination” box it is selecting “Determining by distance”.

D.T.R. (Data Transmission Range) of a neighbour node is set to 250 meters and *D.I.R. (Data Interference Range)* of a neighbour node is set to 500 meters. The rest of parameters are set as below in Figure 4.22.

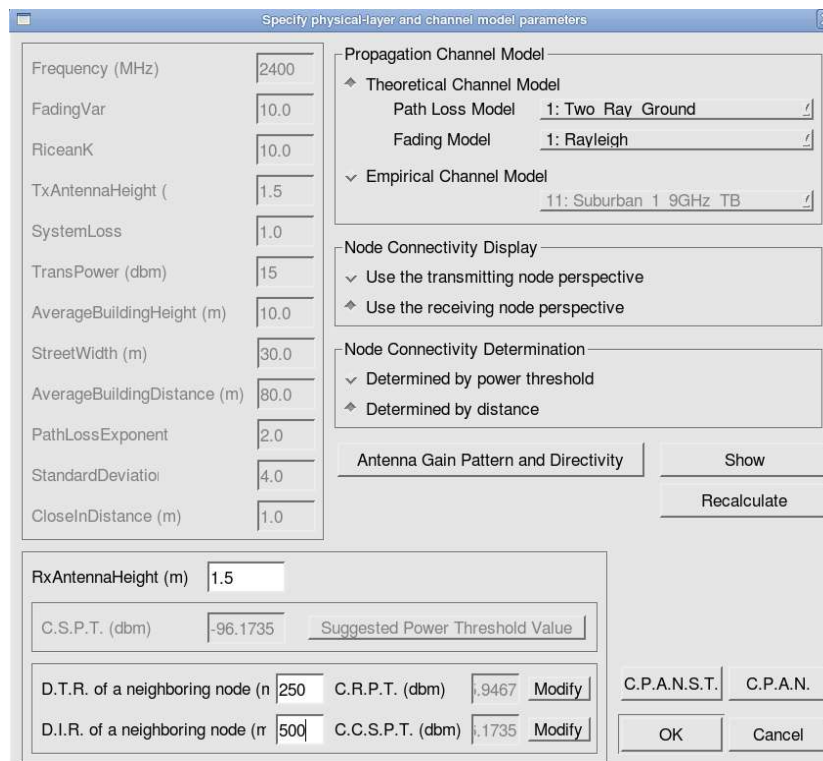


Figure 4.22. Specify physical layer and channel model parameter.

“Recalculate” button has to be clicked before accept all the changes. As we have seen before, if we want to copy this configuration to all the vehicles present in the simulation we have to click the C.P.A.N.S.T. button and “Nodes to Be Selected” > Copy to all nodes of the same network type. As shown in Figure 4.23.

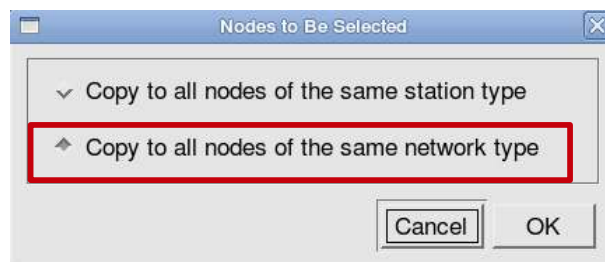


Figure 4.23. Nodes to be selected.

After that, it is opening “G_Setting” > “Simulation” menu to allow editing the simulation parameters such as simulation time and random number seed. Simulation time is set at 200 seconds for all the simulations. And the random number seed is set at 1 because, if the same random number seed is used for a simulation case, the simulation results are repeatable across different runs. As shown in Figure 4.24.

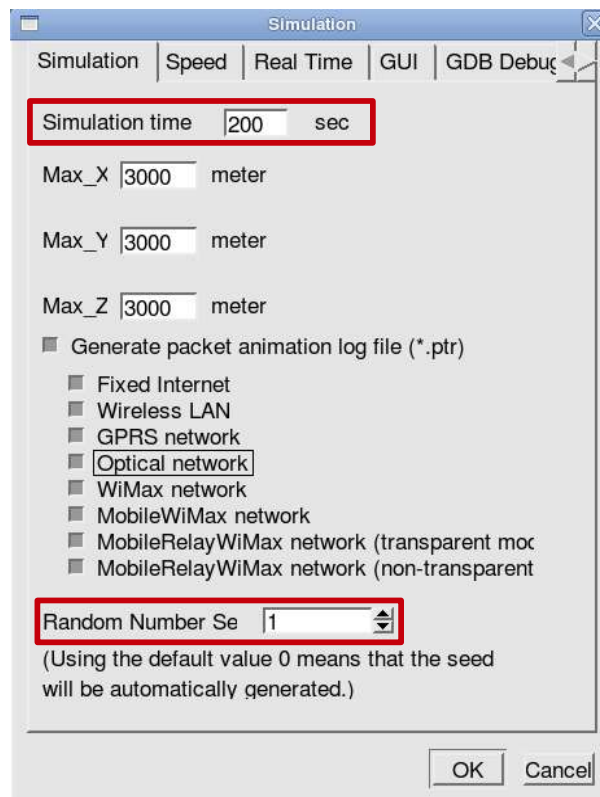


Figure 4.24. Simulation time and random number seed.

The last step in “Edit property” mode, before change to “Run Simulation” mode in order to perform the simulation. We have to configure job dispatcher preference as shown in Figure 4.25. The default port number is 9 800. How it is using a single machine, the IP address can be specified as 127.0.0.1, which is the default IP address that UNIX system automatically assigns to the loopback interface. The user name and password must be valid. For the single machine it is the user account on this local machine.

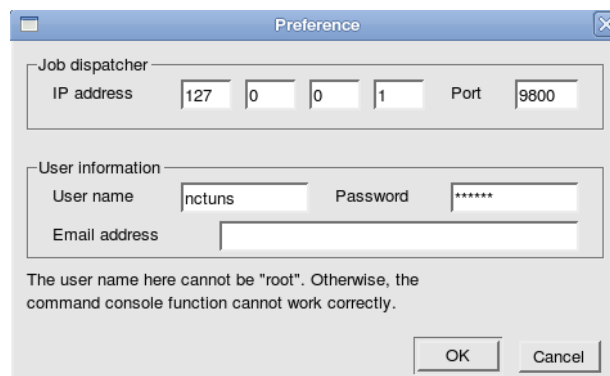


Figure 4.25. Dispatcher configuration.

Now we proceed as described in section 4.2.2.1 to “Run simulation” mode.



Figure 4.26. Run simulation mode.

Once in this mode, the topology is saved and several files are generate. And the simulation could start by selecting the “Simulation” > “Run” option.

After a variable period of time approximately between 30 minutes and 3 hours per each simulation. The following message is presenting on the screen:

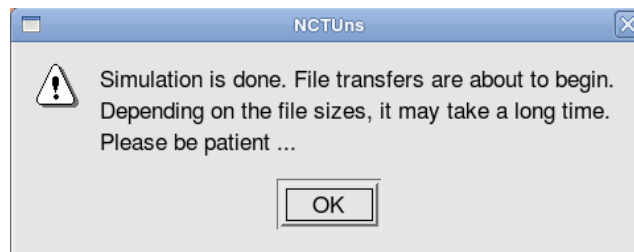


Figure 4.27. Message simulation is done.

We have to accept two dialog boxes and several files will be copied. The simulation is finished. After finishing a simulation, the GUI program will automatically switch itself into the “Play Back” mode and read the packet trace file generated during the simulation. In this mode, it can use the GUI program to replay the packet transmission/reception operation in an animated way. As shown in Figure 4.28.

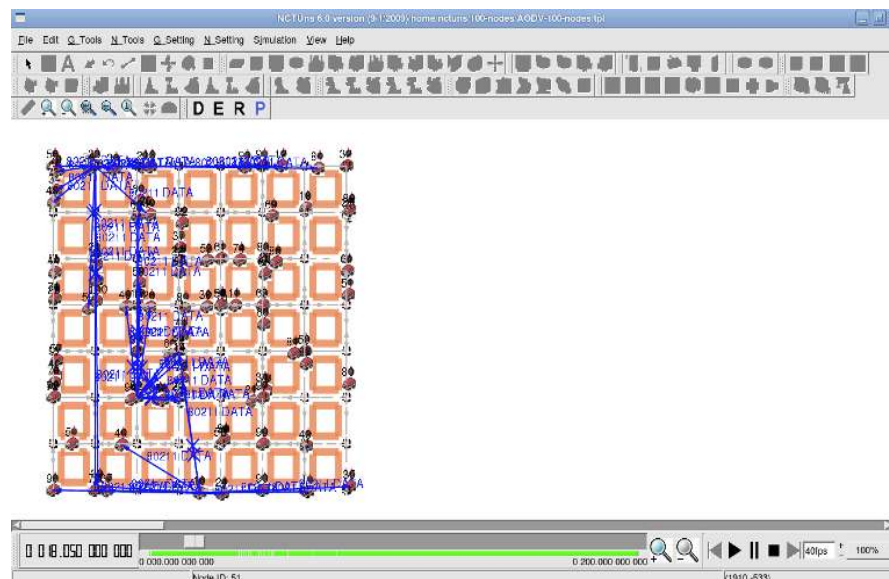


Figure 4.28. Packet animation player.

NCTUns generates a binary trace file, *Simulation_Name.ptr*, which the GUI client reads to do the replays. With the help of the *printPtr* application we can decode the binary file and translate it into a plain text that can be treated more easily. However, we consider that this trace file is not suitable in order to obtain the real source and destination from a packet.

At this way, we decide create a new trace file in order to obtain the required data, the real source and destination from a packet. By adding into the AODV.cc some lines which create the new trace file. For instance:

```
fdtA = fopen("/home/nctuns/Desktop/AODV-tripp","a+"); //tripp
int destino = ipv4addr_to_nodeid(dst_ip); //tripp
int fuente = ipv4addr_to_nodeid(src_ip); //tripp
fprintf(fdtA, "TX_DATA - %lf - %i - %i - %i - %i - %i \n", (double)(GetCurrentTime() *
TICK) / 1000000000.0, get_nid(), fuente, destino, p->pkt_getlen(), p->pkt_getpid()); //tripp
fclose(fdtA); //tripp
```

Traces have the next scheme:

1	2	3	4	5	6	7
TX_DATA	- 130.889699	- 100	- 100	- 51	- 1028	- 82758
RX_DATA	- 131.116812	- 51	- 51	- 100	- 1028	- 82758

The first field is about the type of event TX_DATA (data transmission), RX_DATA (data reception). Second field is the current time of the event in seconds. Third field is current vehicle ID that is transmitting a packet. Fourth field is the real source ID from the packet. Fifth field is real destination ID from the packet. Sixth field is the packet length from the packet. And seventh field is the packet ID.

All these information is treated and filtered with the help of AWK filter, which is a language for massive treatment of data and for processing text files. A file is treated as a sequence of records, any by default each line is a record. Each lines is broken up into a sequence of fields, so we can think of the first work in a line as the first file, the second word has the second field, and so on. An AWK program is of sequence of pattern-action statement. AWK reads the input a line at a time. A line is scanned for each pattern in the program, and for each pattern that matches the associated action is executed. Further information about AWK language is available in [AWK 2011].

On the Annex 1, there is a AWK filter programed for the scenario with 20 vehicles in order to obtain the required data to analyze it. After extract in a plain text file with AWK filter the required data, we can export it to a spreadsheet in order to perform the corresponding data analysis, applied some formulas regarding the metrics to analyze and generate graphics.

The following chapter shows the simulations results obtained with the simulator NCTUns and the Citymob mobility pattern generator regarding to packets losses, delay, throughput and packet delivery. There is a comparison between AODV and AODV-ABE routing protocol in VANETs environments. Where for all the graphics AODV-ABE routing protocol is refer as ABE.

5. SIMULATIONS AND RESULTS

This chapter discusses and analyzes the results obtained from a performance evaluation carried out by using the NCTUns simulator [NCTUns], according to configured parameters in the *GUI (Graphic User Interface)* of NCTUns described on the previous chapter.

5.1 *Simulation Settings*

The most of simulation settings implemented have been described and configured on Chapter 4. Table 5.1 summarizes the parameters involved in the simulations.

We define available bandwidth as the maximum throughput that can be transmitted between end-to-end peers without disrupting any already ongoing transmission in the network. It is crucial to highlight that the required bandwidth established for ABE is 800 kbps. It means, that in real time ABE computes before begin any transmission, an estimation of the available bandwidth for a possible candidate path to forwarding. ABE compares if this estimation of the available bandwidth if it is less than or higher than the requirement. If it is less than 800 kbps, ABE does not forward the packet and discards the path; conversely, if the available bandwidth is higher than 800 kbps, ABE establishes the forwarding path and it transmits the packets.

<i>Parameters</i>	<i>Value</i>
<i>MAC Specification</i>	<i>IEEE 802.11b</i>
<i>Medium Capacity</i>	<i>2 Mbps</i>
<i>Packet Length</i>	<i>1000 bytes</i>
<i>Bit Rate</i>	<i>Constant (CBR)</i>
<i>Data Rate</i>	<i>800 kbps</i>
<i>Traffic</i>	<i>UDP</i>
<i>Grid size (scenario)</i>	<i>1000 m x 1000 m</i>
<i>No. Transmissions</i>	<i>3</i>
<i>Data Transmission Range (DTR)</i>	<i>250 m</i>
<i>Data Interference Range (DIR)</i>	<i>500 m</i>
<i>Path Loss Model</i>	<i>Two Ray Ground</i>
<i>Fading Model</i>	<i>Rayleigh</i>
<i>Simulation Time</i>	<i>200 sec</i>
<i>Maximum Speed</i>	<i>50 km/h (14 m/s)</i>
<i>No. Vehicles</i>	<i>20, 40, 60, 80, 100</i>
<i>Mobility Model</i>	<i>Downtown (Citymob)</i>
<i>Routing Protocol</i>	<i>AODV, ABE</i>

Table 5.1. Simulation Settings.

5.2 Scenario, Simulations and Graphs Description

First of all, we are considered 5 scenarios with 20, 40, 60, 80 and 100 vehicles, respectively. The scenarios are represented in Figures 5.1, 5.8, 5.15, 5.22 and 5.29, which show the distribution of the vehicles into the scenario. It is important to highlight the scenarios only differ in vehicles distribution, mobility pattern and which vehicles are transmitters or receivers. The rest of simulation settings are exactly equal for the 5 scenarios.

In each scenario, it was established three simultaneous transmissions: *Trans 1*, *Trans 2* and *Trans 3*. The vehicles that are transmitting have a green circle and vehicles that are receiving have a red circle, it is depicted also in Figures 5.1, 5.8, 5.15, 5.22 and 5.29. These three simultaneous transmissions are sent a CBR flow from a source to a destination, which they were selected in a random way, each transmission is forwarding packets of 1000 bytes with a data rate of 800 kbps.

It was made in total 100 simulations for the project. We evaluate 5 different urban scenarios with different number of vehicles. That is, per each scenario it was made 20 simulations; it means, 10 simulations for ABE and 10 simulations for AODV per each scenario (5 scenarios), in order to obtain more accurate data to compare the performance for both protocols.

At last, regarding the values represented in the graphs shown along this chapter, they are represented with *Confidence Intervals (CI)* of 95%. A short confidence interval means a short variation between the values; whereas, a long confidence interval means long dispersion of values.

5.3 Metrics Analyzed

As aforementioned, we take into account 5 scenarios in an urban area with different mobility patterns for: 20, 40, 60, 80 and 100 vehicles, respectively. We establish three simultaneous transmissions in each scenario, the objective is to vary the number of vehicles involved to study the effect and impact that causes the higher number of vehicles. Accordingly, we evaluate this effect in terms of: *packet loss ratio, delay, throughput, throughput over time and packet delivery ratio*.

The *Packet Loss Ratio* is calculated by using the following formula:

$$\text{Packet Loss Ratio} = \frac{T_x - R_x}{T_x} \quad (5.1)$$

The difference of the number of transmitted packets (T_x) minus the number of received packets (R_x) divided by the number of transmitted packets (T_x) to obtain the *end-to-end average packet loss ratio*.

The *delay per packet* is calculated by using the following formula:

$$\text{Average delay per packet} = \frac{\sum_{i=1}^{R_x} \text{Delay } i}{R_x} \quad (5.2)$$

The addition of delays means, the sum of the time between a transmitted packet (T_x) was sent, until a received packet (R_x) arrives to destination. This delay is divided by the number of received packets (R_x) to obtain the *end-to-end average delay per packet*.

The *Throughput* is calculated by using the following formula:

$$Throughput = \frac{Rx}{Time} \quad (5.3)$$

To obtain the *end-to-end average throughput* the number of received packets (Rx) are expressed in bits and divided by the total time of simulation (e.g., 200 sec). At the same way, the *end-to-end throughput over time* is calculated. In this case, the throughput is calculated at intervals of 25 seconds.

The *Packet Delivery Ratio* is calculated by using the following formula:

$$Packets \ Delivery \ Ratio = \frac{Rx}{Tx} \quad (5.4)$$

The number of received packets (Rx) is divided by the number of transmitted packets (Tx) in order to obtain the *end-to-end average packet delivery ratio*.

5.4 Scenario with 20 vehicles

It is important to highlight, vehicles movement pattern is random. It is different for each scenario and it is generated by Citymob [Citymob 2008], which is the mobility pattern generator implemented for this project. The vehicles follow the streets, respect the traffic signals and in the intersections can turn right, left or continue in the same direction.

The vehicles distribution in the scenario with 20 vehicles is shown in Figure 5.1, as it is mentioned before, there are three simultaneous transmissions. The vehicles that are transmitting have a green circle and vehicles that are receiving have a red circle. The *Trans 1* is set from vehicle 7 to vehicle 18, the *Trans 2* from vehicle 13 to vehicle 16 and the *Trans 3* from vehicle 1 to vehicle 4 which are sent a CBR at 800 kbps.

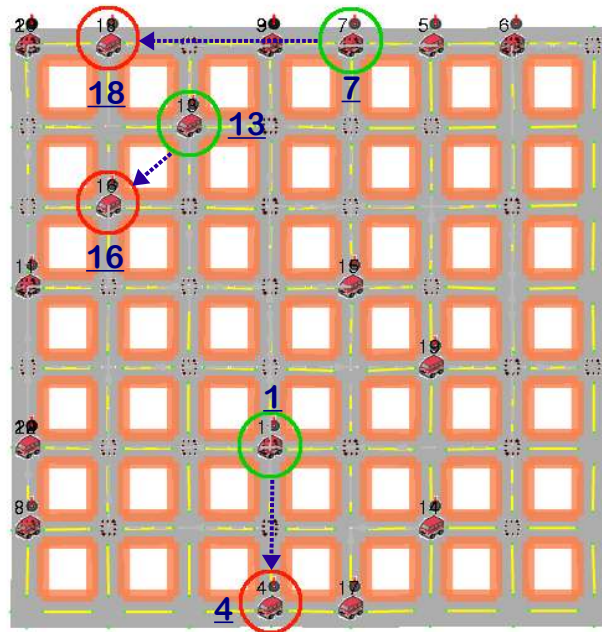


Figure 5.1. Screenshot scenario with 20 vehicles.
Trans 1 (7 to 18). Trans 2 (13 to 16). Trans 3 (1 to 4).

5.4.1 Packet Loss Ratio

The graph in Figure 5.2 which corresponds to scenario with 20 vehicles, it shows the obtained results for this scenario regarding the end-to-end average packet loss ratio. These results were obtained according the formula (5.1) described in section 5.2 and they are taken from the average of 10 simulations per protocol. As it can be seen, AODV results (represented by red bars) in the three transmissions show a packet loss ratio higher than ABE results (represented by blue bars). In the *Trans 1*, AODV suffers a losses of 9.45% higher than ABE; the *Trans 2*, AODV suffers a losses of 12.08% higher than ABE and the *Trans 3*, AODV suffers a losses of 10.67% higher than ABE.

AODV suffers higher losses than ABE in all transmissions because, AODV establishes the forwarding path whenever it finds an end-to-end path without taking into account any requirement, unlike ABE that if it does. Therefore, AODV lost more packets and suffers more packets retransmission. For both protocols, these results are directly dependent on vehicles mobility considering the vehicles population is low and sparse and the possibility to find paths decrease; as a consequence, the possibility to produce paths breakages increases. However, in the *Trans 2* vehicles (13 to 16) remain not so distant and the forwarding paths are more steady. Instead, for the *Trans 1* and *Trans 3* vehicles involved tend to keep away and it is more difficult to find paths to establish the forwarding path.

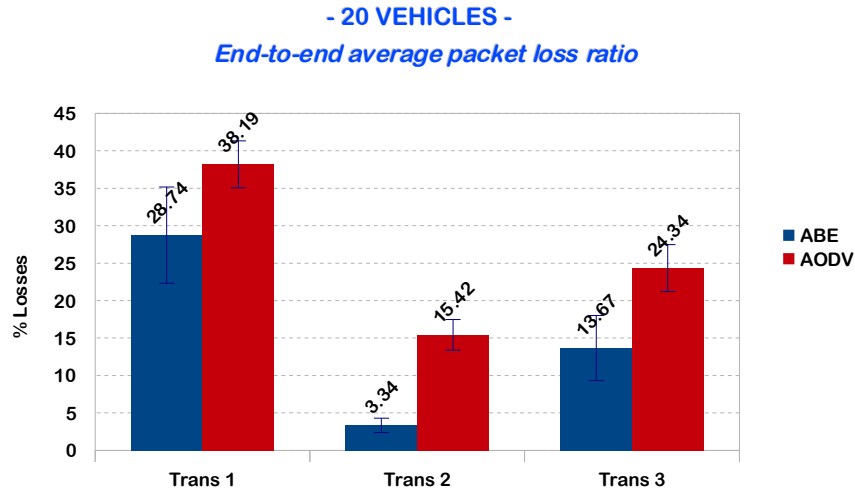


Figure 5.2. End-to-end average packet loss ratio with 20 vehicles.

5.4.2 Average Delay per Packet

Figure 5.3 shows the obtained results regarding the end-to-end average delay per packet with the scenario for 20 vehicles. These results were obtained with the formula (5.2) described in section 5.2, and they are taken from the average of 10 simulations per protocol. AODV delay (represented by red bars) is higher in the three transmissions than ABE delay (represented by blue bars). In this case, in the *Trans 1* the difference is 0.57 seconds higher than ABE, in the *Trans 2* the difference is 11.98 seconds higher than ABE and in the *Trans 3* the difference is 2.40 seconds higher than ABE. Delay results are dependent on the packet loss ratio results. For instance, the delay for ABE in *Trans 2* is the lowest followed by the *Trans 3* and *Trans 1* according the packet loss ratio.

That is, ABE delay is lower than AODV delay because. ABE avoids to forward packets when the path does not satisfies the requirement. At this hand, ABE to some extent prevents packets loss and packets retransmission. Because, the chosen path based on the requirement it will be less vulnerable to suffers delay due the selected path satisfies the require bandwidth. We assume that the path selected has less traffic and congestion hence, the packets does not suffers a high delays due congestion and/or retransmissions.

As a consequence, the delay for ABE is admissible. On the other hand, AODV suffers higher delay than ABE because, the high losses and therefore retransmissions suffered by AODV causes high delay in the transmissions.

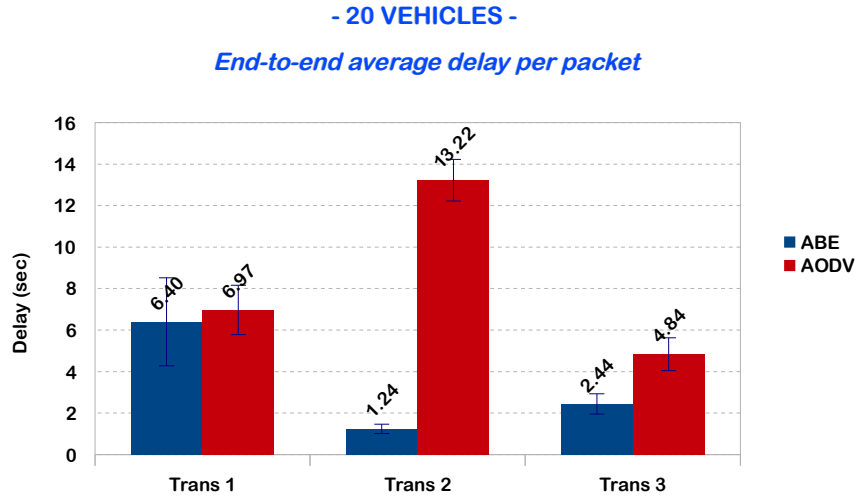


Figure 5.3. End-to-end average delay per packet with 20 vehicles.

5.4.3 Throughput

Table 5.2 shows the number of packets Tx (transmitted), Rx (received) and lost; that were obtained for this scenario with 20 vehicles. It can be observed that in the *Trans 1* and *Trans 3* the number of packets that were forwarded with ABE are less than the number of packets forwarded with AODV but, in the *Trans 2* occurs the opposite, there are more packets forwarded with ABE.

It is important to take into account in this scenario that the vehicle population is low and sparse. Therefore, it is more difficult to establish and maintain the forwarding paths and; as a consequence, this depends directly on the mobility pattern that follows the vehicles. Due the vehicles are moving in different directions along the streets.

For this case, the vehicle movement favors the *Trans 2* (13 to 16), because it is possible to find more paths. Therefore, *Trans 2* is forwarding more packets as it can be seen in Table 5.2 for ABE. This means that, the vehicles 13 and 16 remain not so distant during the simulation goes on, and for the *Trans 1* and *Trans 3* the vehicles can follow directions that keep away.

	Packets	ABE	AODV
Trans 1 (7 to 18)	Tx	3841	14761
	Rx	2739	9131
	Lost	1102	5630
Trans 2 (13 to 16)	Tx	13233	11353
	Rx	12780	9599
	Lost	453	1754
Trans 3 (1 to 4)	Tx	3644	16227
	Rx	3134	12251
	Lost	510	3976

Table 5.2. Number of packets Tx, Rx and lost with 20 vehicles.

Regarding the end-to-end average throughput, it is calculated with the formula (5.3) described in the section 5.2, considering a time of 200 seconds, which is the total simulation time. Also, the throughput results are taken from the average of 10 simulations per protocol.

It can be observed in Figure 5.4, the *Trans 2* shows the highest throughput for ABE, due it is receiving more packets as shown in Table 5.2. The *Trans 1* and *Trans 3* show less throughput compared to AODV. To clarify this case, that AODV shows more throughput than ABE; in the *Trans 1* and *Trans 3*, it is important to take into account the proportion of the number of packets Tx (transmitted), Rx (received) and lost. That is, although AODV is transmitting and receiving more packets it is also losing more packets. If we see the previous Figures 5.2 and 5.3 regarding losses and delay, AODV results are also higher than with ABE results.

As it mentioned on chapter 3 in section 3.4. RREQ is only sent when ABE wants to establish the forwarding path in the route discovery process. Once the path has been established the transmission begins. ABE does not monitor the path to ensure that the path continues to maintaining the bandwidth set. That is, the reason why on graph depicted in Figure 5.4, ABE maintains lower throughput than AODV.

ABE stops the transmission until the path is broken and ABE has to compute another path to reestablish the transmission and then, there is some spent time for ABE to do this process in which ABE does not transmit packets and the throughput decreases. By other hand, AODV only discovers and establish the forwarding end-to-end path hence, AODV does not spend an extra time to compute another path and the throughput is not affected.

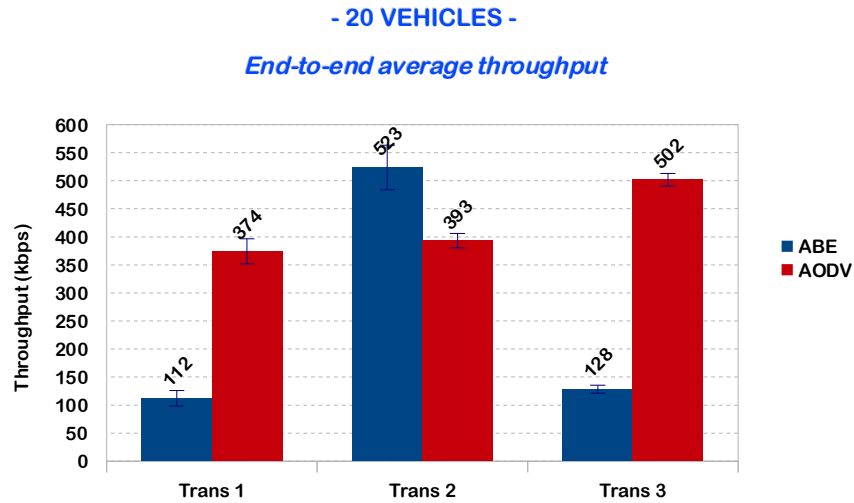


Figure 5.4. End-to-end average throughput with 20 vehicles.

5.4.4 Throughput Over Time

Respecting the throughput over time for this scenario with 20 vehicles, in the graphs depicted in Figures 5.5 and 5.6, they show the corresponding throughput evolution over time obtained for AODV and ABE, respectively. It was calculated at intervals of time of 25 seconds for the three transmissions, according the formula (5.3) described in section 5.2.

The graph in Figure 5.5, AODV shows a throughput over time more or less steady situated between at 350 and 550 kbps but also, AODV provides high losses and high delay, which is not fair to applications that require to guarantee low delay and low packet lost and therefore, lower packets retransmissions.

Whenever that AODV finds an end-to-end path establish and begin to transmit packets without take into account any requirement restriction so when there is a link broken, as soon begins to transmit a end-to-end path is available. As a consequence, the time that AODV takes to restore the transmission is lower than ABE. This is the main reason the transmissions with AODV can achieve greater throughput than ABE because, AODV does not require take extra time to compute the paths. And for ABE there is some spent time in which ABE does not transmit packets and the throughput decreases.

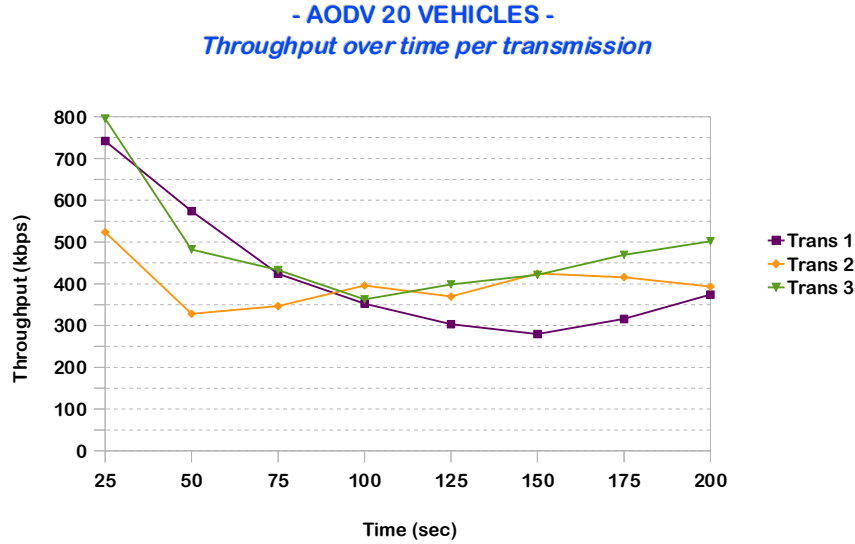


Figure 5.5. Throughput over time with 20 vehicles for AODV.

On the other hand, in Figure 5.6 corresponding the throughput over time for ABE, the *Trans 2* performs better than AODV in this scenario. It maintains a throughput more or less constant between the 350 and 550 kbps also, and even it tends to increase.

In the case of the other two transmissions: *Trans 1* and *Trans 3*, the throughput tends to decrease, because initially the vehicles remain relatively close and as the simulation goes on, the vehicles tend to follow different directions that keep away and therefore, it is difficult to establish the forwarding paths, taking into account that the vehicles population is low and sparse.

Although, the throughput over time for the *Trans 1* and *Trans 3* is lower than AODV. The *Trans 1* and *Trans 3* provide lower losses and lower delay than AODV.

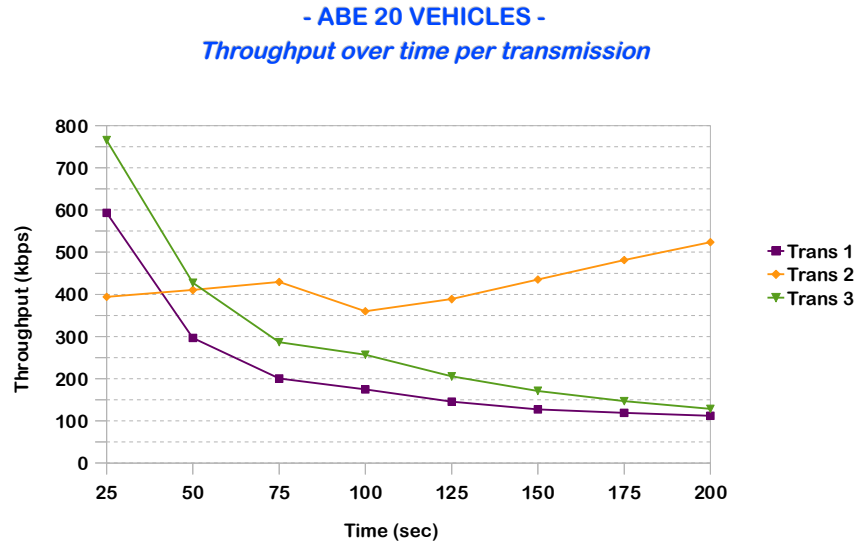


Figure 5.6. Throughput over time with 20 vehicles for ABE.

5.4.5 Packet Delivery Ratio

Finally, for this scenario with 20 vehicles, we show in Figure 5.7 the end-to-end average packet delivery ratio, which shows the packets delivery percentage per transmissions. These results were taken from the average of 10 simulations per protocol with the formula (5.4) describe in section 5.2. ABE results are better than AODV results, as shown in Figure 5.7 where we can see that ABE delivers more packets percentage than AODV.

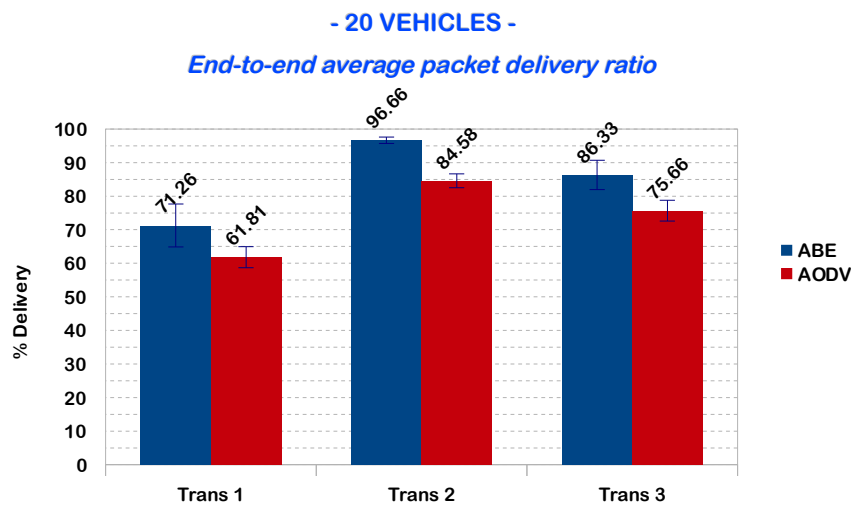


Figure 5.7. End-to-end average packet delivery ratio with 20 vehicles.

5.5 Scenario with 40 vehicles

Vehicles distribution in the scenario with 40 vehicles is shown in Figure 5.8, also there are three simultaneous transmissions and the mobility pattern of vehicles is random. The vehicles that are transmitting have a green circle and vehicles that are receiving have a red circle. The *Trans 1* is set from vehicle 29 to vehicle 8, the *Trans 2* from vehicle 11 to vehicle 35 and the *Trans 3* from vehicle 32 to vehicle 7 which are sent a CBR at 800 kbps.

It is important to underline, that this scenario is different from the previous scenario as to vehicles distribution and also the vehicles that are transmitting and receiving. The rest of simulation settings are exactly equal as previous scenario.

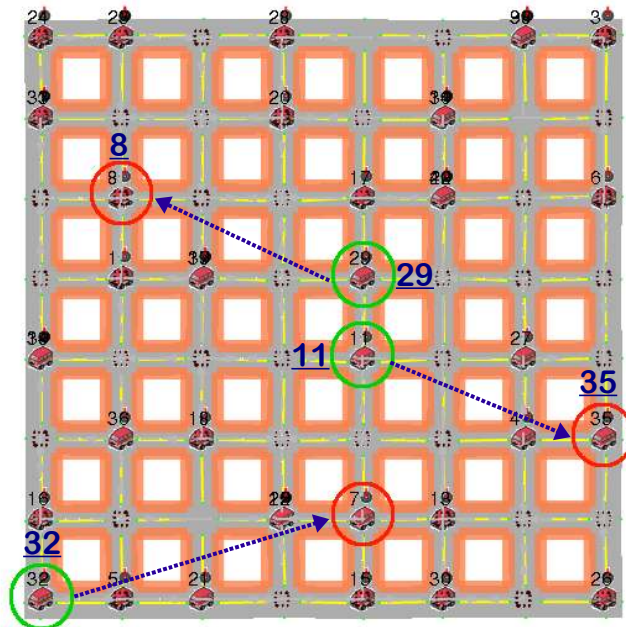


Figure 5.8. Screenshot scenario with 40 vehicles.
Trans 1 (29 to 8). Trans 2 (11 to 35). Trans 3 (32 to 7).

5.5.1 Packet Loss Ratio

The graph in Figure 5.9 which corresponds the scenario with 40 vehicles, it shows the obtained results for this scenario regarding the end-to-end average packet loss ratio. These results were obtained according the formula (5.1) described in section 5.2, and they are taken from the average of 10 simulations per protocol. As it can be seen, the packet lost difference among AODV results (represented by red bars) and ABE results (represented by blue bars) is low in the *Trans 1* and *Trans 2* and high in the *Trans 3*.

In the *Trans 1*, AODV suffers a losses of 1.55% higher than ABE; and the *Trans 2*, AODV suffers a losses of 0.93% higher than ABE; instead, in the *Trans 3* the difference is high, AODV suffers a losses of 10.37% higher than ABE. However, although the difference in the *Trans 1* and *Trans 2* between AODV results and ABE results is low for this particular scenario, at the same way that in the previous scenario regarding the losses, ABE performs better than AODV. In order to clarify this issue is necessary analyze the other metrics.

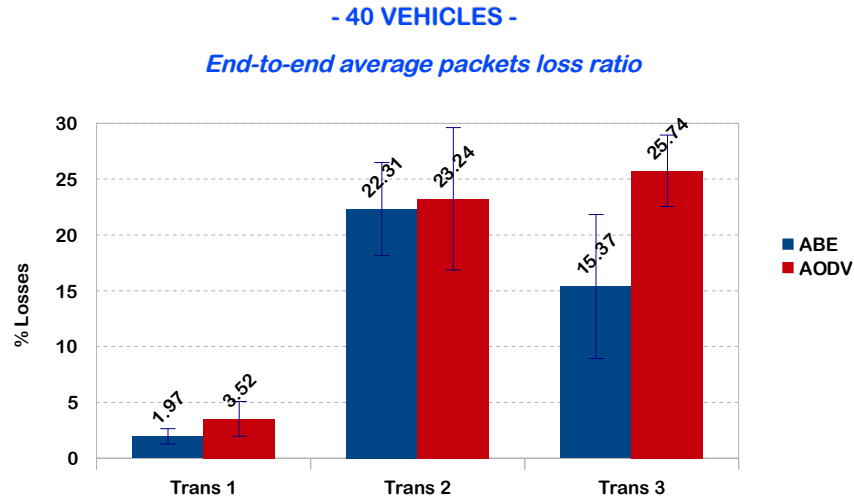


Figure 5.9. End-to-end average packet loss ratio with 40 vehicles.

5.5.2 Average Delay per Packet

Regarding the end-to-end average delay per packet, it can be observed in Figure 5.10, the results obtained for AODV (represented by red bars) are higher than the ABE (represented by blue bars), these results were obtained with the formula (5.2) described in section 5.2, and they are taken from the average of 10 simulations per protocol. In this case, the difference between the *Trans 1* and the *Trans 2* is low. In the *Trans 1*, AODV suffers a delay of 1.02 seconds more than ABE; in the *Trans 2*, AODV suffers a delay of 1.49 seconds more than ABE and in the *Trans 3*, AODV suffers a high difference of 12.07 seconds more than ABE according the end-to-end average delay per packet.

If we observe the behavior of the three transmissions in terms of losses and delay. The three transmissions maintain a similar behavior proportional in function of relation between losses and delays. That is, for ABE the *Trans 1* and *Trans 2* maintain low difference according losses and delay with respect AODV, and for the *Trans 3* maintain a high difference losses and delay with respect AODV also.

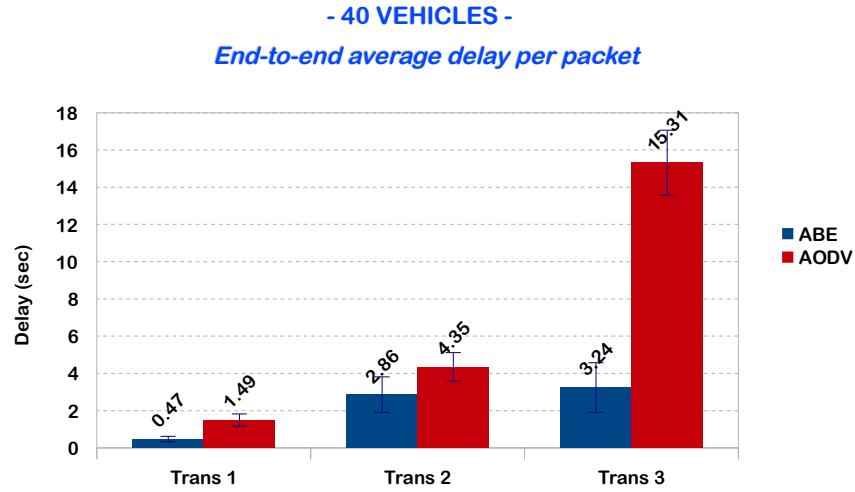


Figure 5.10. End-to-end average delay per packet with 40 vehicles.

5.5.3 Throughput

Table 5.3 shows the number of packets Tx (transmitted), Rx (received) and lost, that were obtained for this scenario with 40 vehicles. It can be observed that the *Trans 1* is the highest for ABE followed by the *Trans 3* and the *Trans 2* in terms of number of packets TX (transmitted). However, the packets Tx (transmitted) by AODV still remains higher with respect to ABE for the three transmissions, because AODV establishes the forwarding path whenever it finds a path and ABE only when the path satisfies the requirement for this reason ABE transmits less packets than AODV.

Packets		ABE	AODV
Trans 1	Tx	16298	16933
	Rx	15975	16336
	Lost	323	597
Trans 2	Tx	4612	16201
	Rx	3527	12491
	Lost	1085	3710
Trans 3	Tx	7934	15354
	Rx	6828	11380
	Lost	1106	3974

Table 5.3. Number of packets Tx, Rx and lost with 40 vehicles.

The end-to-end-average throughput is calculated with the formula (5.3) described in the section 5.2, considering a time of 200 seconds, which is the total simulation time. Also, the throughput results are taken from the average of 10 simulations per protocol.

Regarding the throughput for ABE, we assume that the obtained results depends on the mobility pattern of the vehicles. In this particular scenario, the mobility pattern of the vehicles favors the *Trans 1*, because ABE is finding more paths that satisfies the bandwidth required and it is forwarding higher number of packets than the other two transmissions, and the losses remain low with 1.97% and delay with 0.47 seconds, as it can be seen in Figures 5.9 and 5.10, respectively. As a consequence, the *Trans 1* presents the highest throughput for ABE, as it can be seen in Figure 5.11. On other hand, the *Trans 2*, presents the lowest throughput because it suffers the highest losses with 22.31% as shown in Figure 5.9 for this scenario.

As it can be observed in Figure 5.11 the throughput for AODV is higher than ABE for all the transmissions, because in Table 5.2 shows that AODV is Rx (received) higher number of packets than ABE, but at the expense of suffer high losses and delay, as it can be seen in Figures 5.9 and 5.10, which is not suitable for applications that are sensitive to delays and losses.

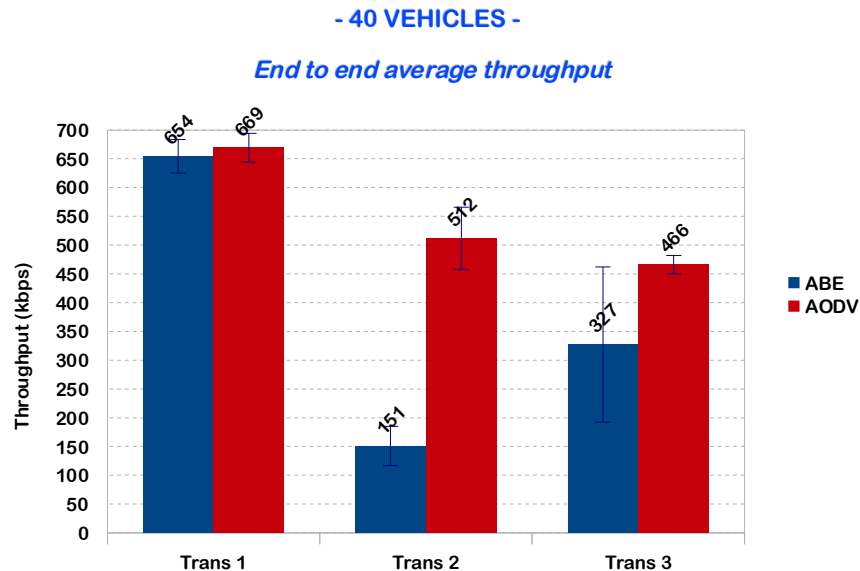


Figure 5.11. End-to-end average throughput with 40 vehicles.

5.5.4 Throughput Over Time

Respecting the throughput over time for this scenario with 40 vehicles, in the graphs depicted in Figures 5.12 and 5.13, they show the corresponding throughput evolution over time obtained for AODV and ABE, respectively. It was calculated at intervals of time of 25 seconds for the three transmissions, according the formula (5.3) described in section 5.2.

As shown in Figure 5.11, the throughput for AODV presents the highest values for the three transmissions; and in Figure 5.12, it tends to follow a constant throughput around the 450 and 700 kbps. Nevertheless, still the AODV has a higher throughput than ABE is at the expense of suffer high delay and high losses.

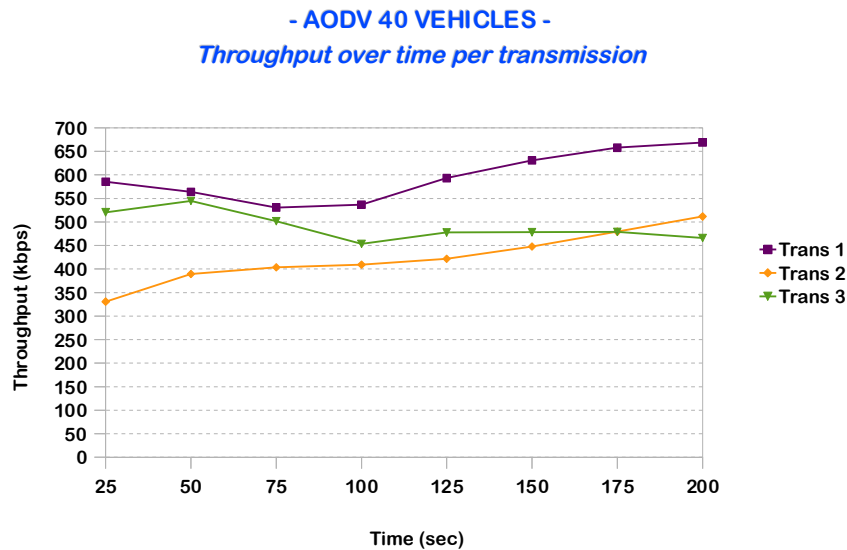


Figure 5.12. Throughput over time with 40 vehicles for AODV.

Relative the end-to-end average throughput over time for ABE, as shown in Figure 5.13, the *Trans 1* performs better than the others two transmissions. *Trans 1* around second 75 suffers a decrease, but after second 75 it tends to increase. Instead, the *Trans 2* and *Trans 3* tend to follow a throughput pattern lower than the initial but, more or less constant and despite this, it provides low losses and low average delay per packet. The increase and/or decrease throughput is consequence on the streets that follow the vehicles which can favors to establish the forwarding paths or not.

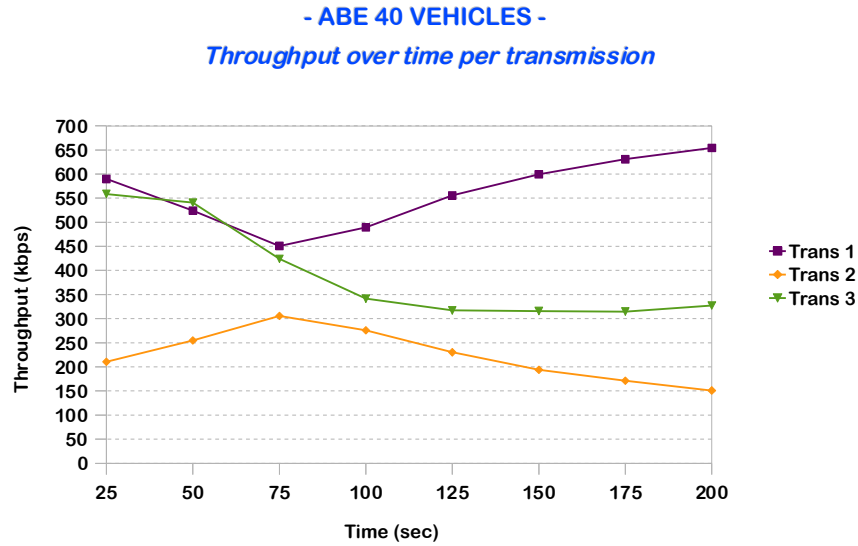


Figure 5.13. Throughput over time with 40 vehicles for ABE.

5.5.5 Packets Delivery Ratio

Finally, for this scenario with 40 vehicles, we show in Figure 5.14, the end-to-end average packet delivery ratio, which shows the packets delivery percentage per transmissions. These results were taken from the average of 10 simulations per protocol with the formula (5.4) describe in section 5.2. ABE results are better than the AODV results, as shown in Figure 5.14 where we can see that ABE delivers more packets percentage than AODV.

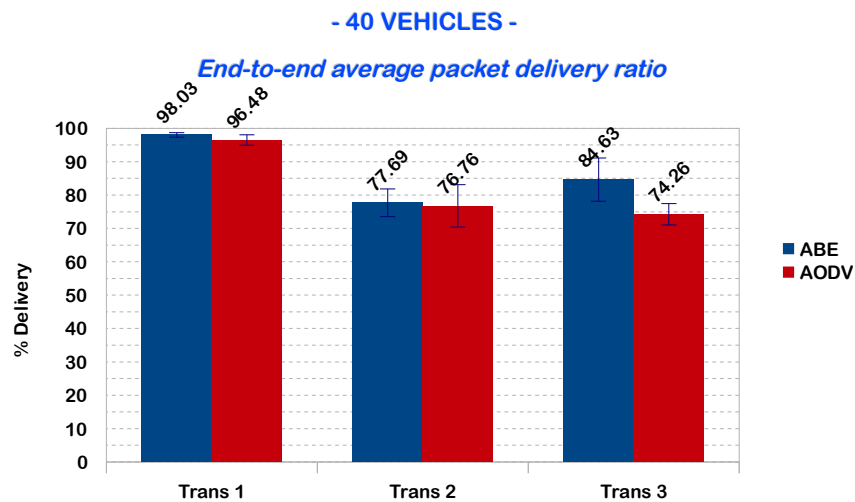


Figure 5.14. End-to-end average packet delivery ratio with 40 vehicles.

As it can be seen, conforms to increase the number of vehicles involved into the scenarios, there are more possibilities for ABE to establish the forwarding paths, but remains to analyze what occurs with the delay, losses and throughput in the following scenarios.

5.6 Scenario with 60 vehicles

This scenario is different from previous two scenarios as to vehicles distribution and also the vehicles that are transmitting and receiving. The vehicles that are transmitting have a green circle and vehicles that are receiving have a red circle. The rest of simulation settings are exactly equal as previous two scenarios. Vehicles follow the streets, respect the traffic signals and in the intersections can turn right, left or continue in the same direction.

Vehicles distribution in the scenario with 60 vehicles is shown in Figure 5.15, at the same way there are three simultaneous transmissions and the mobility pattern is random. The *Trans 1* is set from the vehicle 9 to vehicle 30, the *Trans 2* from vehicle 55 to vehicle 10 and the *Trans 3* from vehicle 27 to vehicle 44 which are sent a CBR at 800 kbps.

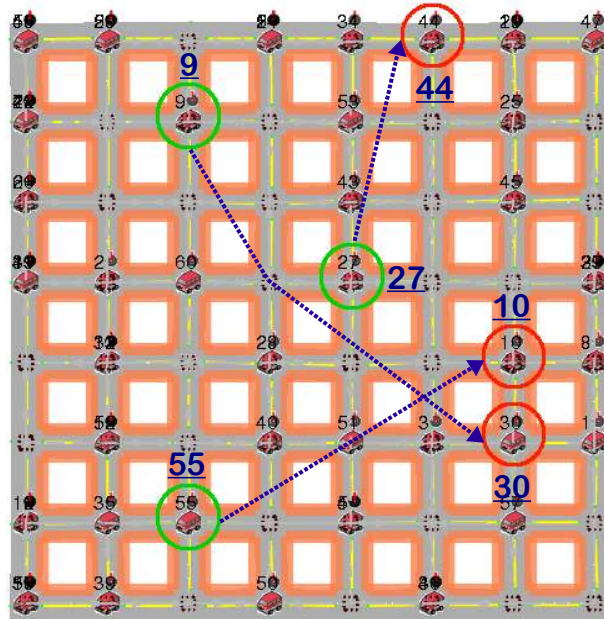


Figure 5.15. Screenshot scenario with 60 vehicles.
Trans 1 (9 to 30). Trans 2 (55 to 10). Trans 3 (27 to 44).

5.6.1 Packets Loss Ratio

The graph in Figure 5.16 which corresponds the scenario with 60 vehicles, it shows the obtained results for this scenario regarding the end-to-end average packet loss ratio. These results were obtained according the formula (5.1) described in section 5.2, and they are taken from the average of 10 simulations per protocol. As in the previous two scenarios, the difference according packet loss ratio among AODV results (represented by red bars) and ABE results (represented by blue bars) is denoted too. ABE results take the advantage over AODV results and the difference is more marked that in the previous two scenarios. In the *Trans 1*, AODV suffers a difference of 23.3% higher than ABE; the *Trans 2*, AODV suffers a difference of 8.20% higher than ABE; and the *Trans 3*, AODV suffers a difference of 10.39% higher than ABE.

AODV needs end-to-end paths for data forwarding, which it is difficult to handle because, the end-to-end paths break frequently due the high mobility of vehicles. Despite this, ABE uses the basic operation of AODV to find paths. Nevertheless, ABE includes in its operation an algorithm to improve the performance based on estimate the available bandwidth for a wireless link. Based on that parameter, ABE decides if the link satisfies the bandwidth required for the application or not. If the requirement is satisfied, ABE establish the forwarding path, otherwise it does not establish the forwarding path. For this reason, the losses for ABE are lower than AODV, because ABE takes the forwarding decision based on this parameter. Unlike AODV that forwards a packet whenever it finds a path to a particular destination.

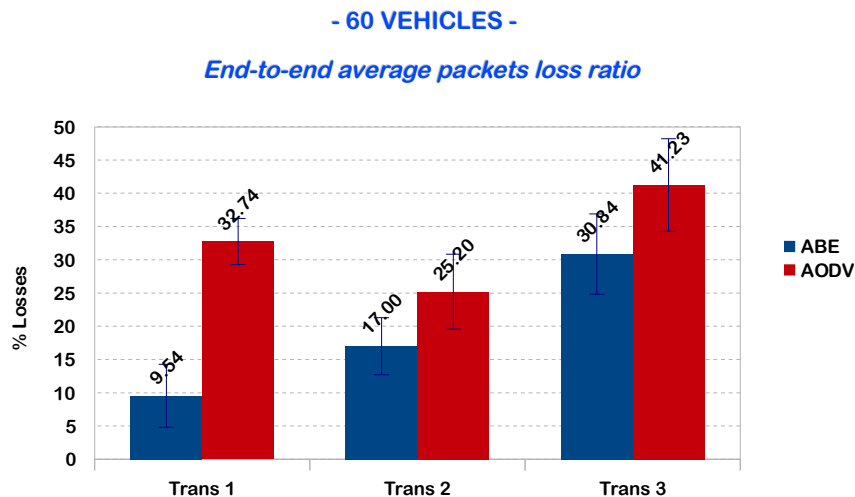


Figure 5.16. End-to-end average packet loss ratio with 60 vehicles.

5.6.2 Average Delay per Packet

In relation the delay in Figure 5.17 for this scenario with 60 vehicles, it shows the obtained results regarding the end-to-end average delay per packet. These results were obtained with the formula (5.2) described in section 5.2, and they are taken from the average of 10 simulations per protocol. AODV delay (represented by red bars) is higher in the three transmissions than the ABE delay (represented by blue bars).

ABE results shows a lower end-to-end average delay per packet for the three transmissions than AODV. The *Trans 1* suffers a delay difference of 3.85 seconds, the *Trans 2* suffer at 3.96 seconds and the *Trans 3* suffers at 9.55 seconds per packet more than ABE. For this specific scenario, ABE presents the best results regarding the delay compared with the four scenarios analyzed in this project.

That is due, ABE selects the path that satisfies the requirement hence, it selects paths that have less traffic and/or congestion. Therefore, ABE provides lower delay than AODV which selects whenever it finds an end-to-end path.

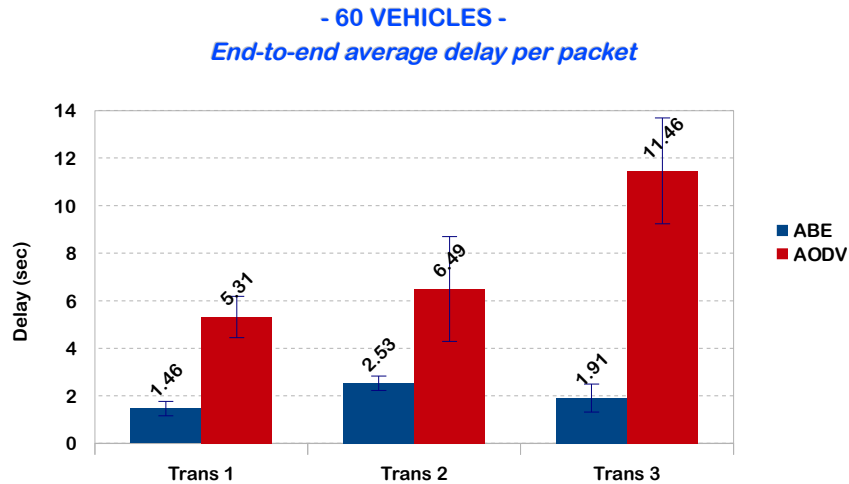


Figure 5.17. End-to-end average delay per packet with 60 vehicles.

5.6.3 Throughput

Table 5.4 shows the number of packets Tx (transmitted), Rx (received) and lost, that were obtained for this scenario with 60 vehicles. As it can be seen, ABE is forwarding less number of packets than AODV nevertheless, if we consider the proportion of number of received packets (Rx) in function of the number of transmitted packets (Tx), the packets lost still remains lower for ABE as shown in Figure 5.16 regarding the packet loss ratio.

	Packets	ABE	AODV
Trans 1	Tx	8466	15603
	Rx	7723	10481
	Lost	743	5122
Trans 2	Tx	6543	15246
	Rx	5422	11461
	Lost	1121	3785
Trans 3	Tx	4475	13685
	Rx	2990	7934
	Lost	1485	5751

Table 5.4. Number of packets Tx, Rx and lost with 60 vehicles.

Regarding the throughput, it is calculated with the formula (5.3) described in section 5.2, considering a time of 200 seconds, which is the total simulation time. Also, the throughput results are taken from the average of 10 simulations per protocol. As it can be observed in Figure 5.18 the end-to-end average throughput. AODV presents higher throughput than ABE.

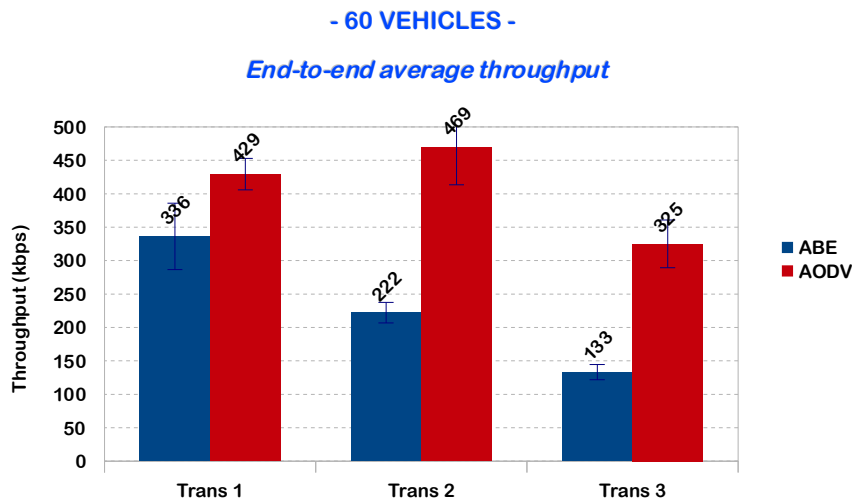


Figure 5.18. End-to-end average throughput with 60 vehicles.

The point is that, ABE uses the basic operation to find paths at the same way that AODV does. For this reason, as it can be seen in Table 5.4, the number of packets transmitted by AODV are the highest, because AODV forwards a packet whenever it finds an end-to-end path, instead ABE only establishes the forwarding paths, when these paths satisfy the minimum bandwidth that requires.

5.6.4 Throughput Over Time

Respecting the throughput over time for this scenario, on the graphs depicted in Figures 5.19 and 5.20, they show the corresponding throughput evolution over time obtained for AODV and ABE, respectively. It was calculated at intervals of time of 25 seconds for the three transmissions, according to the formula (5.3) described in section 5.2.

The graph in Figure 5.19, AODV around the second 75 experiences a decrease and around from the second 150 it maintains a throughput located between the 300 and 500 kbps. However, AODV continues to maintain a high level of losses, delay and therefore, packets retransmissions also. The AODV throughput only is affected when the path is broken and has to establish another end-to-end path.

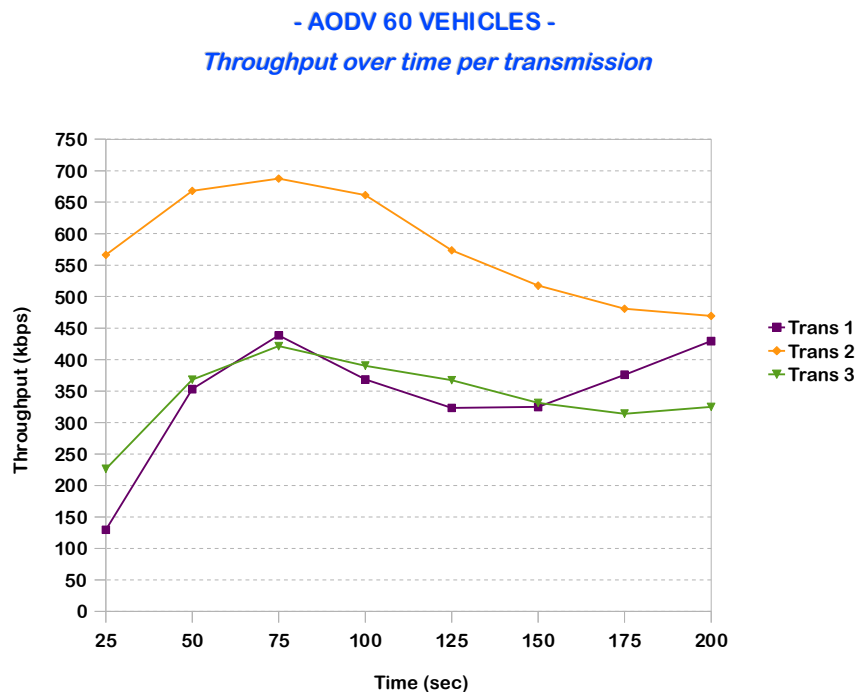


Figure 5.19. Throughput over time with 60 vehicles for AODV.

On the other hand, in Figure 5.20 corresponding the throughput over time for ABE, also at the same way that AODV. ABE at the second 75 experience a throughput decrease, due the effect mobility. Since the vehicles follow the streets and they can randomly in the intersections turn right, turn left or continue at the same direction, respect signals and move quickly, producing paths breakages. It forced to find and evaluate new paths to reestablish the transmissions.

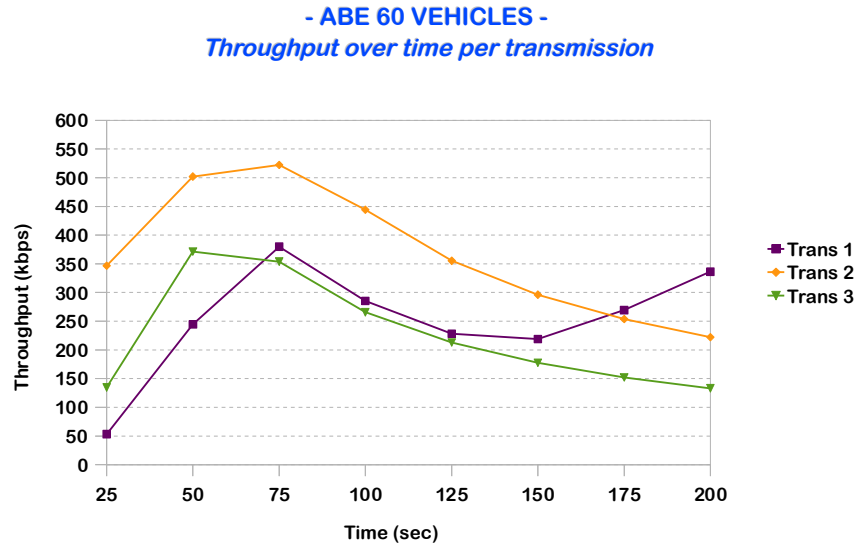


Figure 5.20. Throughput over time with 60 vehicles for ABE.

5.6.5 Packets Delivery Ratio

Finally, for this scenario with 60 vehicles, at the same way that in the previous three scenarios. We show in Figure 5.21 the end-to-end average packet delivery ratio, which shows the packets delivery percentage per transmissions. These results were taken from the average of 10 simulations per protocol with the formula (5.4) describe in section 5.2. ABE results performs always better than AODV results because ABE delivers more packets percentage than AODV.

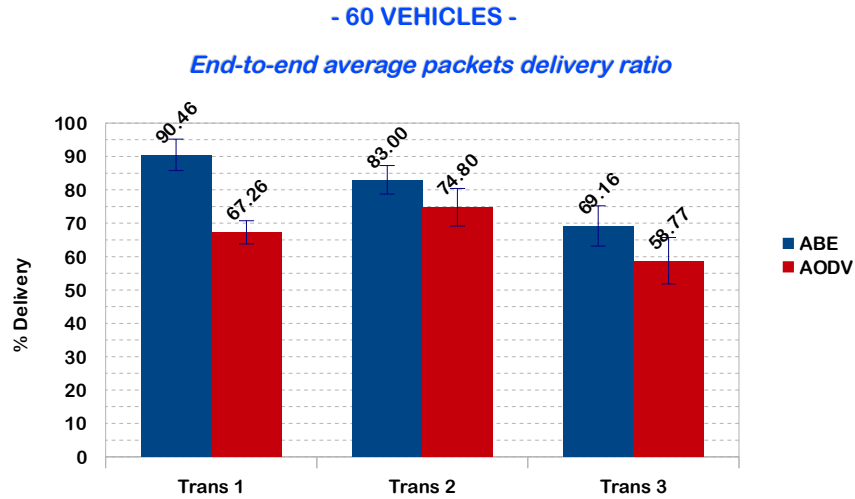


Figure 5.21. End-to-end average packet delivery ratio with 60 vehicles.

5.7 Scenario with 80 vehicles

Vehicles distribution in the scenario with 80 vehicles is shown in Figure 5.22, at the same way there are three simultaneous transmissions and the mobility pattern is random. The vehicles that are transmitting have a green circle and vehicles that are receiving have a red circle. The *Trans 1* is set from vehicle 48 to vehicle 53, the *Trans 2* from vehicle 62 to vehicle 15 and the *Trans 3* from vehicle 31 to vehicle 26 which are sent a CBR at 800 kbps.

Vehicles follow the streets, respect the traffic signals and in the intersections can turn right, left or continue in the same direction. This scenario is different from previous three scenarios as to vehicles distribution and also the vehicles that are transmitting and receiving. The rest of simulation settings are exactly equal as previous three scenarios.

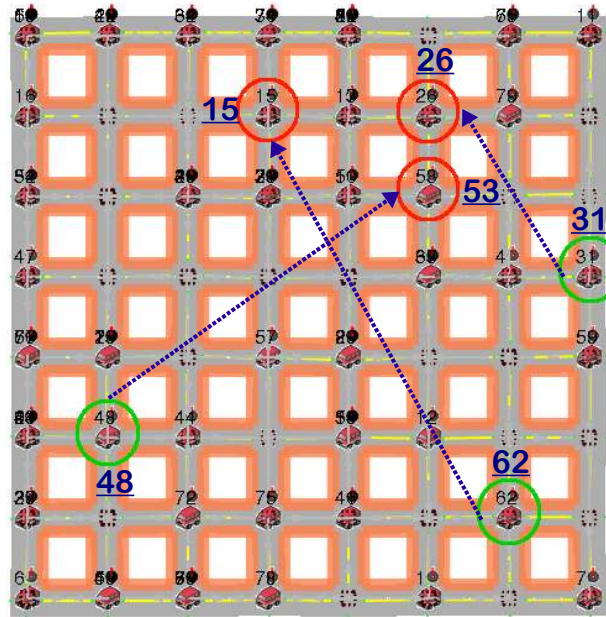


Figure 5.22. Screenshot scenario with 80 vehicles.
 Trans 1 (48 to 53). Trans 2 (62 to 15). Trans 3 (31 to 26).

5.7.1 Packets Loss Ratio

The graph in Figure 5.23 which corresponds the scenario with 80 vehicles, it shows the obtained results for this scenario regarding the end-to-end average packet loss ratio. These results were obtained according the formula (5.1) described in section 5.2, and they are taken from the average of 10 simulations per protocol. As it can be seen, the AODV results (represented by red bars) and ABE result (represented by blue bars). In the *Trans 1*, AODV suffers a losses of 3.58% higher than ABE; the *Trans 2*, AODV suffers a losses of 0.39% higher than ABE; and the *Trans 3*, AODV suffers a losses of 4% higher than ABE. For this particular scenario, ABE results for the *Trans 3*, ABE suffers a 31.58% of losses, this is the worst value obtained for ABE because is the highest packet lost percentage suffering for ABE among all the transmissions for all the scenarios.

Initially, vehicles 31 and 26 are a little close but, conform they are moving during the simulation goes on, they follow directions that keep away and it is more difficult to find paths that satisfies the requirement. Besides that, there is traffic from the other two simultaneous transmissions that can occur collisions, although ABE take into account this factor to compute the forwarding path, this issue can not prevent completely. Also, the fading effect is present, due to shadowing from obstacles.

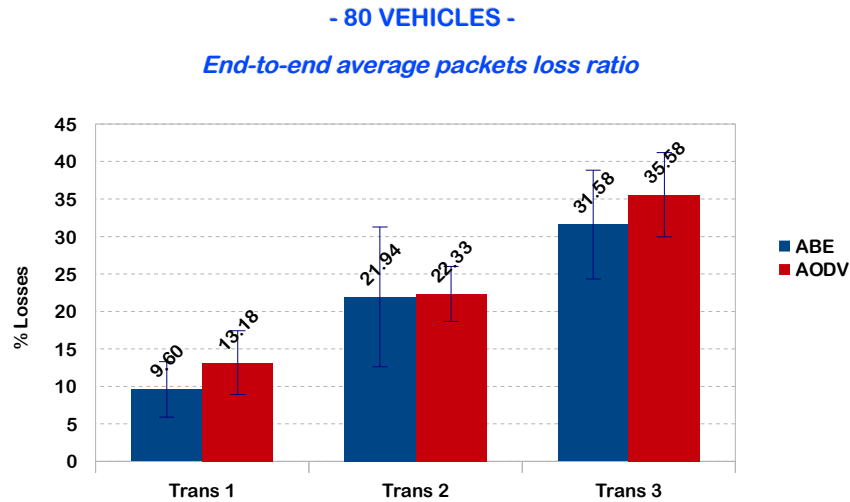


Figure 5.23. End-to-end average packet loss ratio with 80 vehicles.

5.7.2 Average Delay per Packet

Regarding the delay in Figure 5.24 for this scenario with 80 vehicles, it shows the obtained results regarding the end-to-end average delay per packet. These results were obtained with the formula (5.2) described in section 5.2, and they are taken from the average of 10 simulations per protocol. The AODV delay (represented by red bars) is higher in the three transmissions than the ABE delay (represented by blue bars). Differences are for the *Trans 1*, AODV suffers at 10.03 seconds higher than ABE; the *Trans 2*, AODV suffers at 8.01 seconds higher than ABE; and the *Trans 3*, AODV suffers at 14.57 seconds higher than ABE.

The *Trans 3* suffers the poorest performance regarding losses and throughput among all the transmissions in all the scenarios. The number of packets that is Tx (transmitted) and Rx (received) is the lowest as it can be seen in Table 5.5. Therefore, *Trans 3* suffers the highest losses percentage as shown in Figure 5.23. However, taken into account the number of packets transmitted and lost in function of delay of received packets it is not high for this transmission. Because, although the packets losses percentage is high, those packets lost not cause delay in the packets that if it is received.

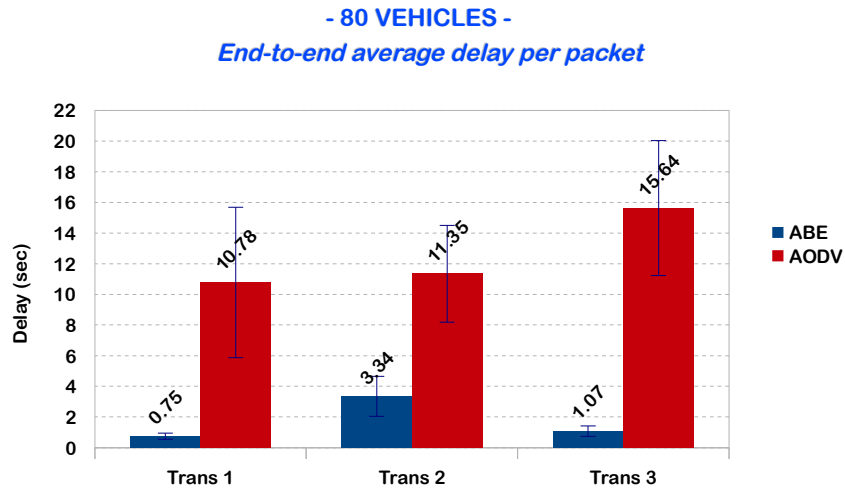


Figure 5.24. End-to-end average delay per packet with 80 vehicles.

5.7.3 Throughput

Table 5.5 shows the number of packets Tx (transmitted), Rx (received) and lost, that were obtained for this scenario with 80 vehicles, there are difference between the packets forwarding for ABE and AODV. For instance in the *Trans 3*, it can be seen for ABE is forwarding 1763 packets and it is losing 31.58% of the transmitted packets and the average delay per packet is 1.07 seconds. On other hand, for this *Trans 3*, AODV is forwarding 14159 packets and it is losing 35.58% and the average delay per packet is 15.64 seconds.

The poorest performance for *Trans 3* for ABE is given in terms of losses and throughput, but delay performs well in comparison with the AODV delay where the difference among the ABE delay and AODV delay is 14.57 higher AODV. This high difference in AODV delay is attributed the high number of hops that the packet has to traverse in order to reach its destination. Because for AODV does not matter the network conditions only matter establish the end-to-end forwarding path.

	Packets	ABE	AODV
Trans 1	Tx	12301	12716
	Rx	11176	11186
	Lost	1125	1530
Trans 2	Tx	4447	13799
	Rx	3778	10707
	Lost	669	3092
Trans 3	Tx	1763	14159
	Rx	1214	9234
	Lost	549	4925

Table 5.5. Number of packets Tx, Rx and lost with 80 vehicles.

The end-to-end-average throughput is calculated with the formula (5.3) described in section 5.2, considering a time of 200 seconds, which is the total simulation time. Also, the throughput results are taken from the average of 10 simulations per protocol.

Regarding the throughput as a direct consequence of losses results, in the graph depicted in Figure 5.25 the throughput for the *Trans 1* is the highest and performance better than the three transmissions, because the packets lost is not so high, instead the *Trans 3* suffers the lowest performance according the throughput.

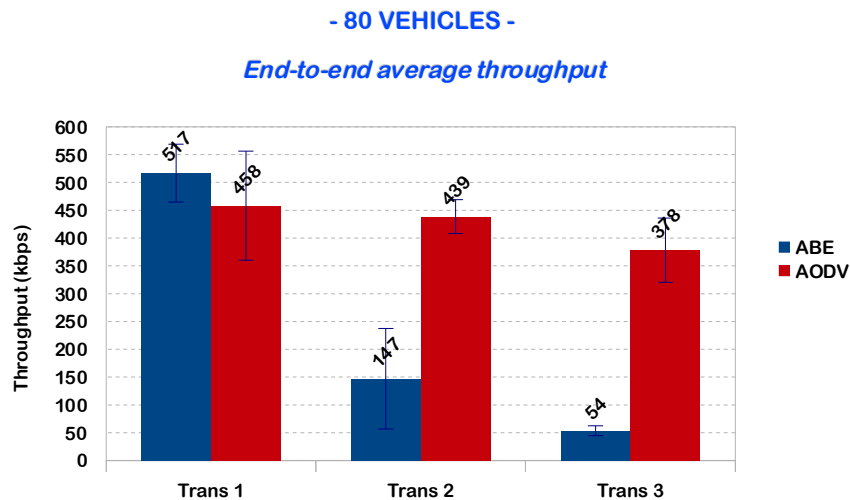


Figure 5.25. End-to-end average throughput with 80 vehicles.

5.7.4 Throughput Over Time

Respecting the throughput over time for this scenario with 80 vehicles, in the graphs depicted in Figures 5.26 and 5.27, they show the corresponding throughput evolution over time obtained for AODV and ABE, respectively. It was calculated at intervals of time of 25 seconds for the three transmissions, according the formula (5.3) described in section 5.2.

The throughput over time for AODV in this scenario from the second 0 to 125 tends to suffer an unsteady throughput due the high mobility, instability and links breakages however, at second 125 the throughput tends to follow a throughput more or less constant between the 350 and 500 kbps, at expenses of suffer high end-to-end average delay per packet.

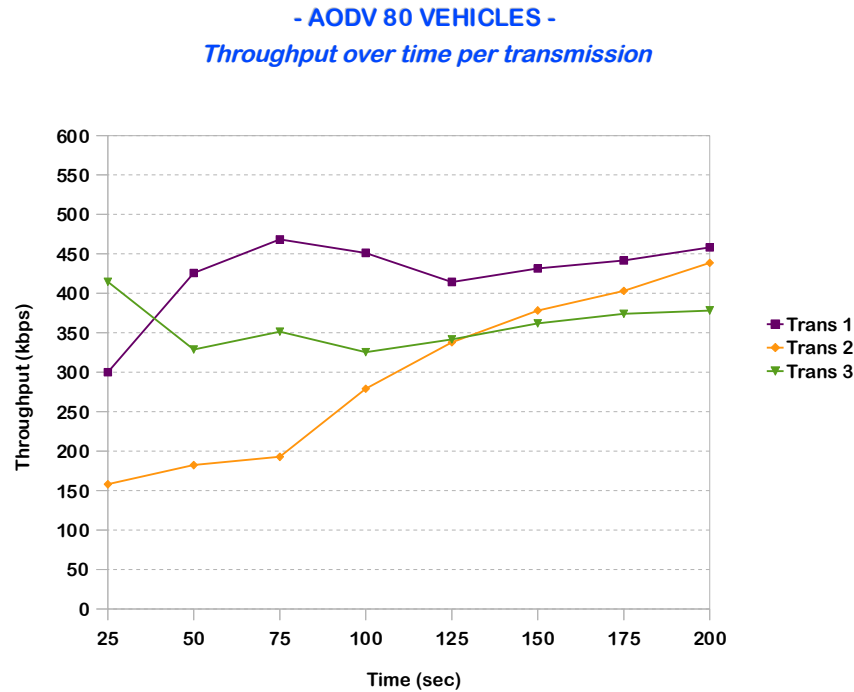


Figure 5.26. Throughput over time with 80 vehicles for AODV.

In Figure 5.27 regarding the throughput over time for ABE as it can be seen, the *Trans 1* from the second 100 it begins to increase significantly, because it is finding more paths to forwards the packets with the available bandwidth that satisfies the requirement. Around the second 75, the *Trans 2* presents an slight increase steady between the 100 and 150 kbps and also it tends to increase.

Regarding the *Trans 3* which perform worst in this scenario, because it presents high losses and low throughput, this last is due to initially the vehicles 31 and 26 are relatively close, and it conforms the simulation goes on it tends to keep away. Therefore, it is difficult to find paths that satisfies the requirement to establish the forwarding paths.

According these results we can assume, that although the vehicles increases is favorable to have more possible options for establishing paths to forwarding the packets, but the mobility factor issue increases also and thereby the possibility to produce paths breakages due the fast mobility factor.

ABE establishes the forwarding path and begins the transmission when received the corresponding RREP, as it is described on chapter 3 in section 3.4. However, taking into account that this is a highly mobile environment, a few seconds or rather well a few thousandths of seconds from established the transmissions and due vehicles movements, the paths conditions can already not be the same affecting throughput.

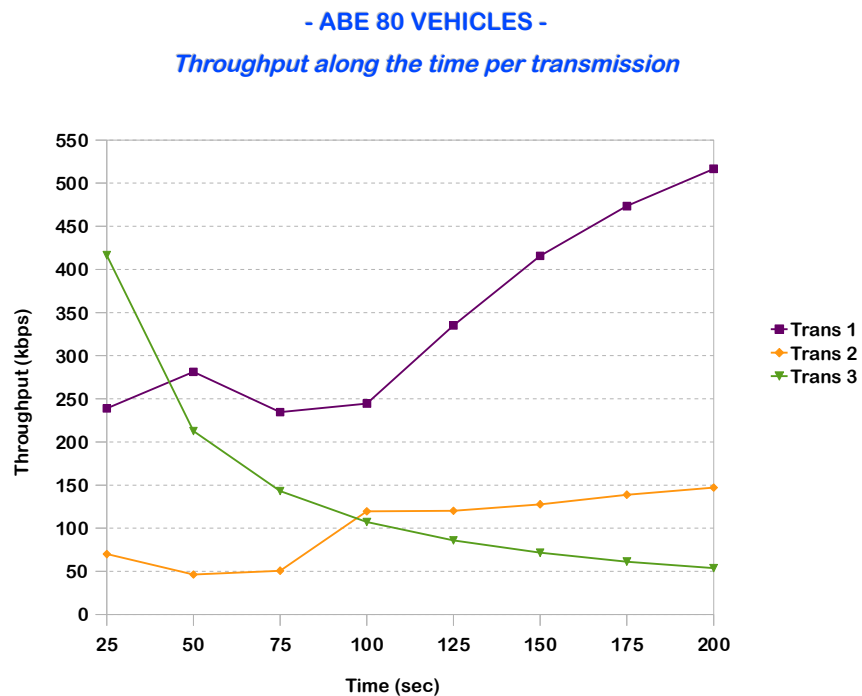


Figure 5.27. Throughput over time with 80 vehicles for ABE.

This issue can be controlled by monitoring continuously the path established and control the transmissions. By adding a congestion control, that when the path does not satisfy the requirement, the transmission is reject and looking for another path. Which is a purpose to implement in ABE for future works. This purpose is explain latter on section 6.2 future works.

5.7.5 Packet Delivery Ratio

Finally, for this scenario with 80 vehicles we show in Figure 5.28 the end-to-end average packet delivery ratio, which shows the packets delivery percentage per transmissions. These results were taken from the average of 10 simulations per protocol with the formula (5.4) describe in the section 5.2. ABE results are better that the AODV results, as shown in Fig. 5.28 where we can see that ABE delivers more packets percentage than AODV.

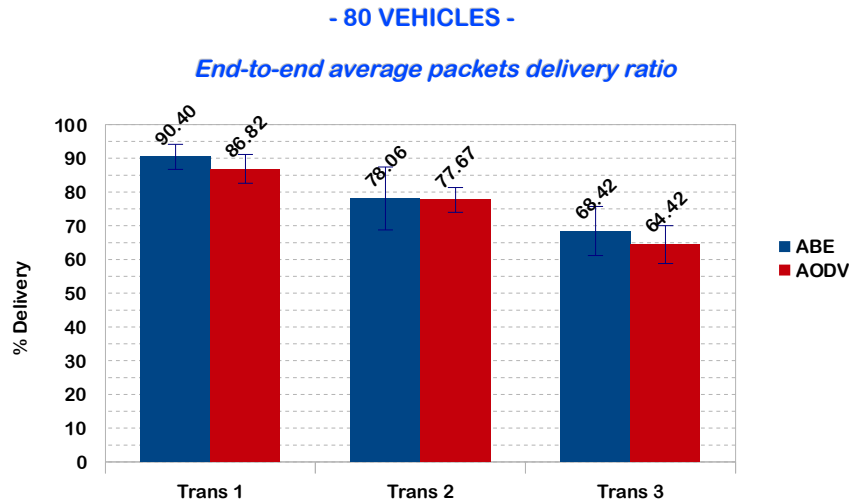


Figure 5.28. End-to-end average packet delivery ratio with 80 vehicles.

5.8 Scenario with 100 vehicles

Vehicles distribution in the scenario with 100 vehicles is shown in Figure 5.29 at the same way that in the previous four scenarios, there are three simultaneous transmissions and the mobility pattern is random. The vehicles that are transmitting have a green circle and vehicles that are receiving have a red circle.

The *Trans 1* is set from vehicle 100 to vehicle 51, the *Trans 2* from vehicle 45 to vehicle 81 and the *Trans 3* from vehicle 2 to vehicle 70 which are sent a CBR at 800 kbps. The vehicles follow the streets, respect the traffic signals and in the intersections can turn right, left or continue in the same direction. The rest of simulation settings are exactly equal as the previous four scenarios.

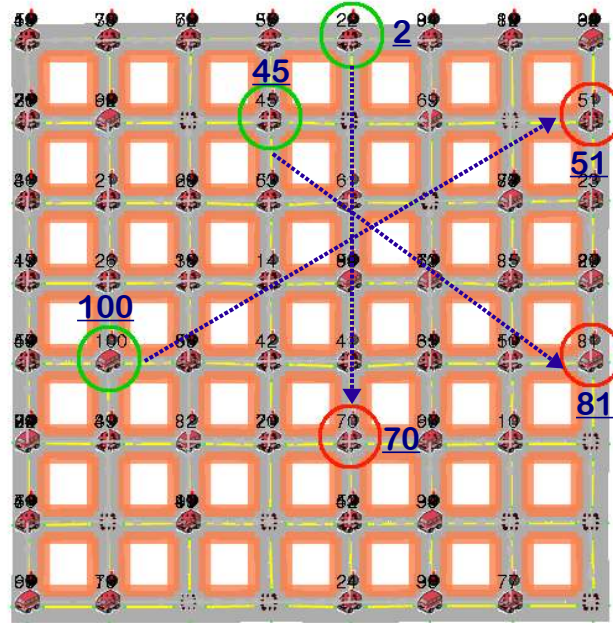


Figure 5.29. Screenshot scenario with 100 vehicles.
Trans 1 (100 to 51). Trans 2 (45 to 81). Trans 3 (2 to 70).

5.8.1 Packets Loss Ratio

The graph in Figure 5.30 which corresponds the scenario with 100 vehicles, it shows the obtained results for this scenario regarding the end-to-end average packet loss ratio. As it can be seen, at the same way that in the previous four scenarios, ABE results maintains a lower end-to-end average packet loss ratio compare with the AODV results, which always remains above ABE. AODV results (represented by red bars) and ABE results (represented by blue bars) were obtained also according the formula (5.1) described in section 5.2, and also they are taken from the average of 10 simulations per protocol in order to obtain more accurate data.

In this last scenario with 100 vehicles; in the *Trans 1*, AODV suffers a losses of 30.79% higher than ABE; the *Trans 2*, AODV suffers a losses of 2.03% higher than ABE; and the *Trans 3*, AODV suffers a losses of 9.77% higher than ABE.

At the same way that in the previous analyzed scenarios, the AODV packet loss ratio is higher than ABE. Because, AODV does not take into account any network consideration and only establish the forwarding path whenever it finds an end-to-end path to a destination, also the high mobility of vehicles increase the losses because it can produces path breakages. Instead, ABE is slightly steady because it includes in its operation an algorithm to improve the performance based on estimate the available bandwidth for a wireless link in order to take the forwarding decision however, ABE suffers the same mobility problem as AODV, since they do not take into account the mobility issue.

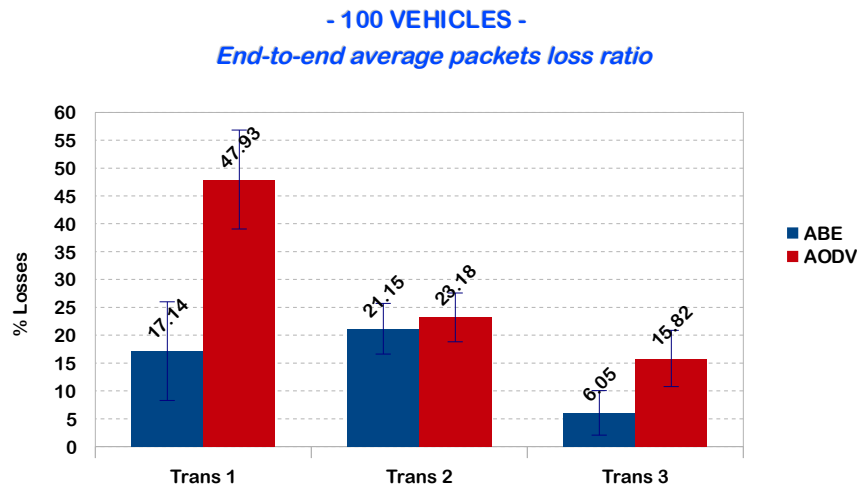


Figure 5.30. End-to-end average packet loss ratio with 100 vehicles.

5.8.2 Average Delay per Packet

Figure 5.31 shows the obtained results regarding the end-to-end average delay per packet with the scenario for 100 vehicles. These results were obtained with the formula (5.2) described in section 5.2, and they are taken from the average of 10 simulations per protocol. AODV delay (represented by red bars) and ABE delay (represented by blue bars).

At the same way, that in the others four previous scenarios analyzed. ABE results maintains a lower end-to-end average delay per packet, compared with the AODV results which always remains above ABE. At this particular scenario, the *Trans 1* suffers 12.56 seconds, the *Trans 2* suffers 2.39 seconds and the *Trans 3* suffers 3.81 seconds more for AODV than ABE.

We can see that under the moderate mobility and number of vehicles in urban scenarios, ABE performs satisfactorily better than AODV, as it can see in this scenario with 100 vehicles. Where the packet loss ratio is keeping below 22% as shown Figure 5.29. And also the delay is keeping below 5 seconds per packet.

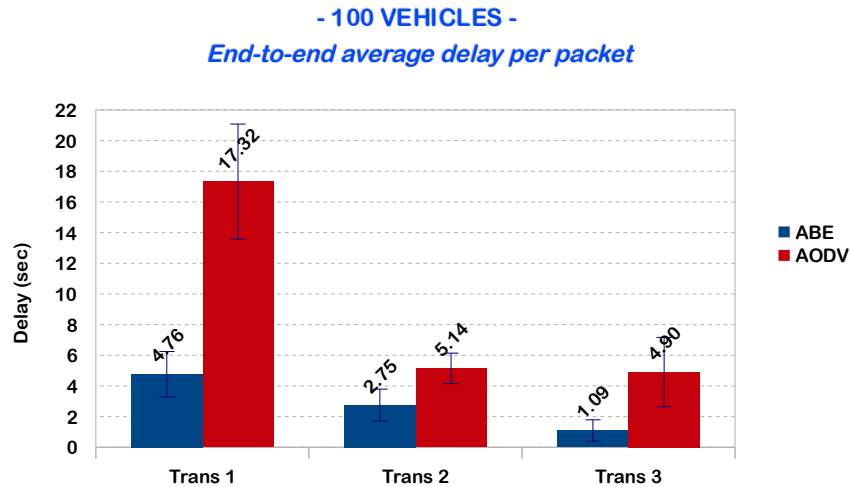


Figure 5.31. End-to-end average delay per packet with 100 vehicles.

5.8.3 Throughput

Table 5.6 shows at the same way that in the previous four scenarios analyzed, the number of packets Tx (transmitted), Rx (received) and lost; that were obtained for this scenario. It can be observed that in the packets Tx (transmitted) by AODV are higher than those packets Tx (transmitted) by ABE. Because AODV has the nature to forward a packet whenever it finds a path to a destination, instead ABE only forwards a packet when the available bandwidth satisfies the requirement.

	Packets	ABE	AODV
Trans 1	Tx	6701	12818
	Rx	5604	6619
	Lost	1097	6199
Trans 2	Tx	5845	14520
	Rx	4735	11130
	Lost	1110	3390
Trans 3	Tx	10462	14438
	Rx	9982	12143
	Lost	480	2295

Table 5.6. Number of packets Tx, Rx and lost with 100 vehicles.

Regarding the end-to-end average throughput, it is calculated with the formula (5.3) described in section 5.2, considering a time of 200 seconds, which is the total simulation time. Also, the throughput results are taken from the average of 10 simulations per protocol.

For this scenario with 100 vehicles, it can be observed in Figure 5.4, regarding the end-to-end average throughput, *Trans 1* experiences a little difference between AODV and ABE be higher for ABE. And the *Trans 3*, although AODV is higher than ABE the difference is not longer. This is due the effect of vehicles mobility and therefore the frequent link breakages.

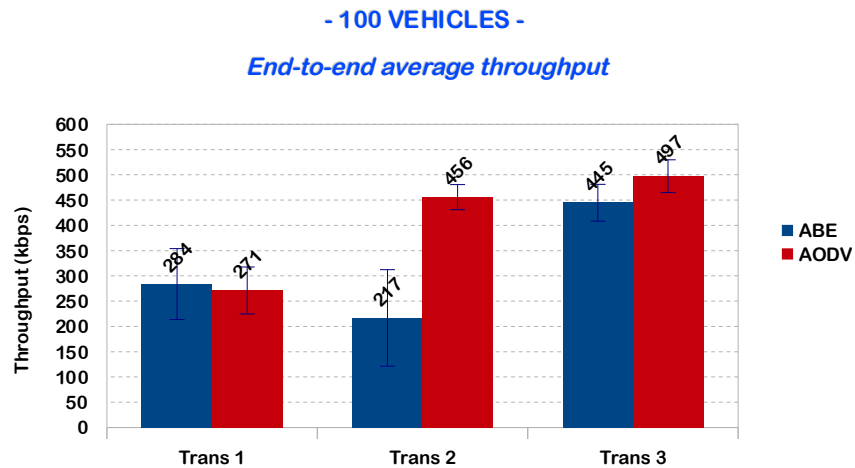


Figure 5.32. End-to-end average throughput with 100 vehicles.

Based on these results it indicated that conforms increase the number of vehicles is possible to find more useful paths for ABE. Because, ABE always is looking for paths that satisfies the required bandwidth to forwarding packets. In order to this, if ABE finds more paths, it forwards more packets. Thereby increase the throughput, moreover always maintaining a low level of losses an delay as shown the graphs regarding losses and delay from all the previous scenarios analyzed. However, it is always depends on the vehicles mobility which is an issue for routing challenge in VANETs.

5.8.4 Throughput Over Time

Respecting the throughput over time for this scenario with 100 vehicles, in the graphs depicted in Figures 5.33 and 5.34, they show the corresponding throughput evolution over time obtained for AODV and ABE, respectively. It was calculated at intervals of time of 25 seconds for the three transmissions, according the formula (5.3) described in section 5.2.

As it can be seen in Figure 5.33 and as shown in the previous graphs analyzed regarding the throughput over time for AODV. In general, throughput for AODV tends to increase because having more vehicles the possibility to find more paths also increases hence, against to obtain high level of throughput in AODV. The transmissions suffers high delay and losses, which is not feasible to applications that require low delay and losses.

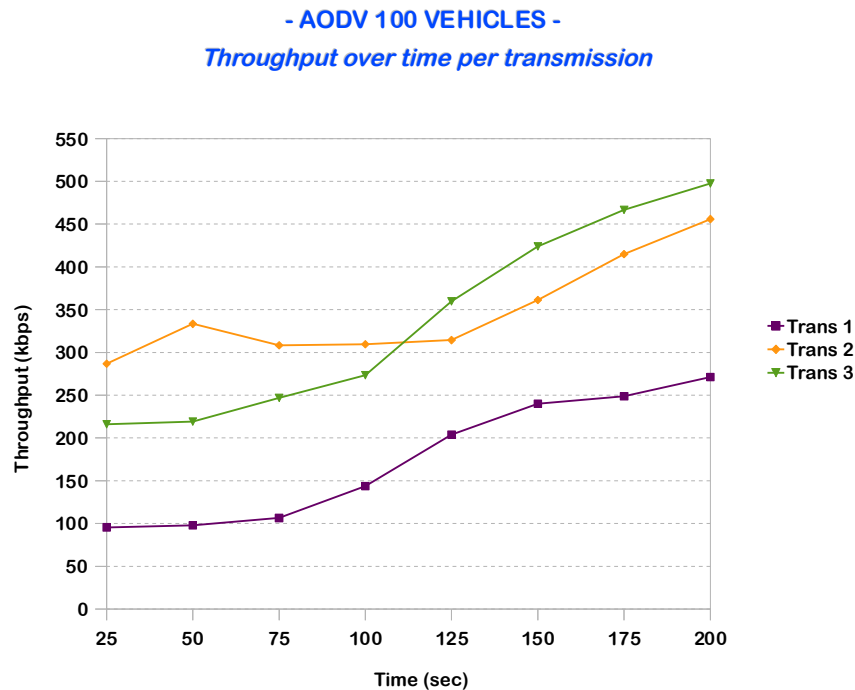


Figure 5.33. Throughput over time with 100 vehicles for AODV.

On the other hand, in Figure 5.34 corresponding the throughput over time for ABE, for this case, the *Trans 1* and *Trans 3* around the second 75, both transmissions experiences a noticeable increase of throughput, and *Trans 2* shows a throughput more or less steady between the 200 and 300 kbps. Also, the three transmissions provides a low level of delay and losses.

For this scenario, all the vehicles involved in the three transmissions: *Trans 1* (vehicle 100 to vehicle 51), *Trans 2* (vehicle 45 to vehicle 81) and *Trans 3* (vehicle 2 to vehicle 70) they are not relatively close. And conforms the simulation goes on, vehicles from the *Trans 1* and *Trans 3* tend to bring near hence, the possibility to find more paths increases, also the transmitting packets and this increases the throughput.

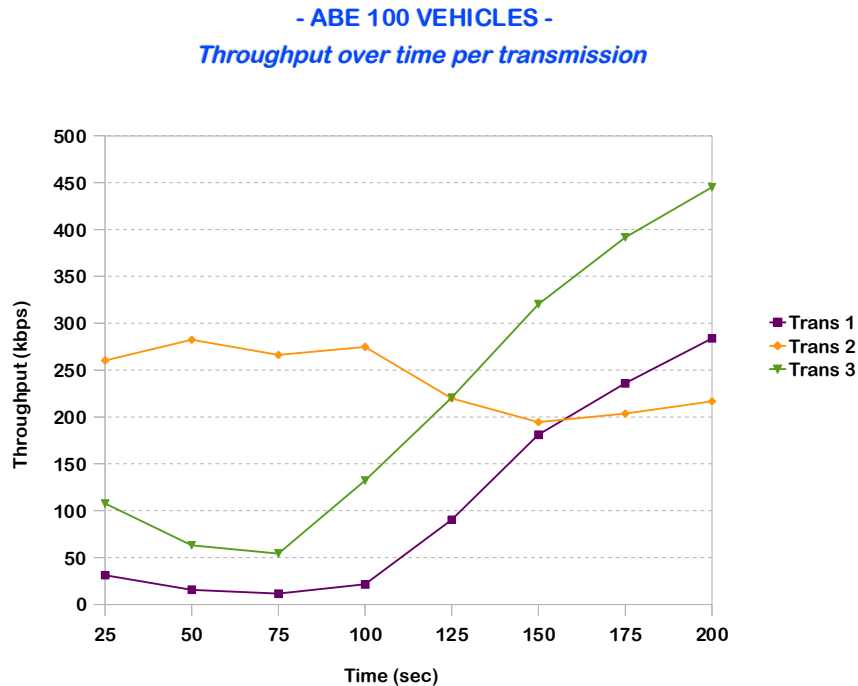


Figure 5.34. Throughput over time with 100 vehicles for ABE.

5.8.5 Packets Delivery Ratio

Finally, for this scenario with 100 vehicles, we show in Figure 5.35 the end-to-end average packet delivery ratio, which shows the packets delivery percentage per transmissions. These results were taken from the average of 10 simulations per protocol with the formula (5.4) describe in section 5.2. ABE results (represented by blue bars) are better than AODV results (represented by red bars) as shown in Fig. 5.35 where we can see that ABE delivers more packets percentage than AODV.

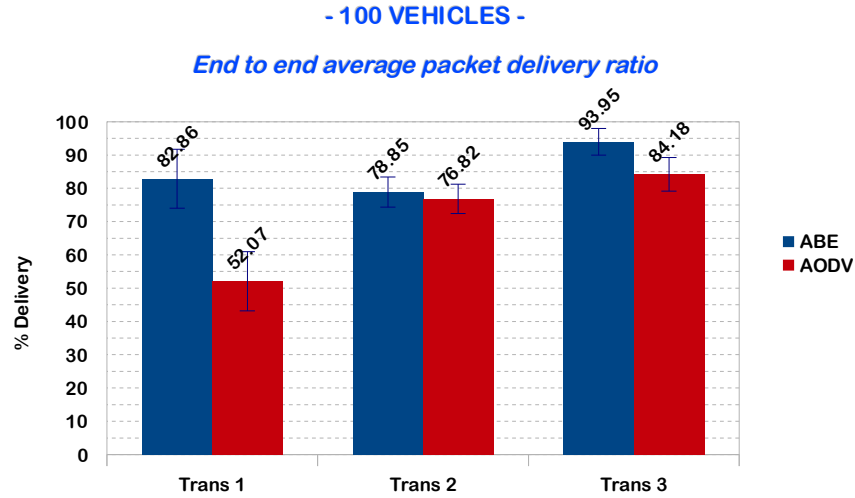


Figure 5.35. End-to-end packet delivery ratio with 100 vehicles.

Accordingly, all these results carried out it shows that AODV and ABE suffer the same problems regarding high mobility, since they do not take into account the mobility issue. Nonetheless, we can see for all the scenarios analyzed that ABE performance better regarding losses and delay which is suitable for some multimedia applications that require a to guarantee minimum bandwidth, short delay and low packets loss. However, all conclusions that we obtained from the scenarios analyzed are described and discussed in detail in the next chapter and also the future works that may arise.

6. CONCLUSIONS AND FUTURE WORK

6.1 *Conclusions*

The main objective of this Master Thesis was to carry out a performance evaluation and comparison between AODV and ABE routing protocols in vehicular urban scenarios.

The main conclusions that we have reached are the following:

- ABE establishes the forwarding path only if this path satisfies the bandwidth requirement of the source. We use the RREQ messages to check that condition during the route discovery process. However, on route maintenance process ABE does not take any action. That is, when at some point on the transmission ABE fails to satisfy the required bandwidth, ABE does not do anything until the path is broken. And again the route discovery process to establish a new forwarding path begins.
- ABE losses perform lower than AODV losses. The reason is that ABE establishes the forwarding path only if the path satisfies the bandwidth requirement; conversely, AODV establishes the

forwarding path whenever it finds an end-to-end path. At this hand, ABE avoids to transmit a packet when the path not satisfies the bandwidth requirement. Hence, at some extent ABE prevents packets losses and packet retransmissions. On the other hand, AODV transmits a packet whenever it finds a path to a destination and AODV does not take into account any network condition so the packets losses are higher than using ABE.

- ABE delay performs lower than AODV delay. Because ABE selects the forwarding path based on the requirement and at some extent ABE selects paths that will be less vulnerable to suffer delay due to the selected path has less traffic and congestion. Hence, the packets do not suffer high delays due to congestion and/or retransmissions.
- ABE throughput perform lower than the achieved AODV throughput. This is because ABE spends a little more time to prepare the path because of the calculation to estimate the available bandwidth. And there is little time that the transmissions are inactive, therefore the throughput decreases. On the other hand, when AODV finds an end-to-end path it forwards the packets, which it does not need to do any extra calculation or spend time. For this, AODV throughput is higher than ABE because the transmissions do not remain inactive.
- As we can see, as the number of vehicles increase it is possible to find more useful paths for ABE. Therefore, we can assume that ABE performs better than AODV in environments with moderate mobility and moderate number of vehicles.
- The great benefit from ABE is that by estimating the available bandwidth and guaranteeing the required bandwidth on the established forwarding paths, ABE guarantees low packet loss ratio, low delay per packet and hence the packet delivery ratio is high. ABE provides QoS to transmissions that are sensitive to delay and losses.
- The big factor that always has influence in the performance of any routing protocol in VANETs, is the high mobility of vehicles. This is a factor unpredictable and non-measurable.

6.2 Future Work

Some of the future works that may arise are the following:

- Implement a congestion control in ABE. That is, during the transmissions remain established, congestion control continuously monitor the available bandwidth in the established path on route maintenance process. As proposed in [Salem 2006] where the authors provide a solution to guarantee end-to-end for real time traffic in ad hoc networks which is based on *PDS model (Proportional Differentiated Services)* aims to achieve better performance for high priority class to low priority class within fixed pre-specified quality spacing. They implement a *Congestion Control (CC)* that when a new flow arrives and there is not enough capacity in the network. The only solution is the rejection of one or more flows in order to re-establish the QoS of the remaining flows in the network. The CC selects a victim flow from class k to be rejected, according to rejection parameters that are associated to classes in a proportion manner. The higher the priority, the smaller the rejection parameters.
- With the congestion control included in ABE, perform simulations to evaluate urban scenarios comparing ABE to other routing protocols proposed for VANETs which offer QoS. Principally, geographic-based protocols which are the most suitable to operate in VANETS environments.
- Other important issue to be consider in VANET communications is the privacy of senders' information, especially in types of service where certain level of senders' anonymity is required like report of road infractions. Further analysis of this topic is required to find out the balance between performance communication and anonymity level. This issue is tackled by another student currently.

ANNEX 1. AWK FILTER

```

BEGIN{
    output1 = "LT-1-ABE_20_.txt"      #Output file
    output2 = "LT-2-ABE_20_.txt"      #Output file
    output3 = "LT-3-ABE_20_.txt"      #Output file

    send_data1A=0      #Add the transmitted packets
    recv_data1A=0      #Add the received packets
    end_RX1A=0         #Add the time of the received packets
    start_TX1A=0       #Store the time of the transmitted packets
    total_start_TX1A=0 #Add the time of the transmitted packets
}

{
    packet=$1          #Kind of packet
    yo=$5              #Transmitting node
    source=$7           #Initial source of the packet
    dst_final=$9        #Final destination of the packet
    pkt_len=$11         #Packet length
    time=$3             #Current time
    pkt_id=$13          #Packet ID

# ----- Transmission from node 7 to 18 -----

    if (packet == "TX_DATA" && yo == "7" && source == "7" && dst_final == "18" && pkt_len
== "1028" && (time < 25))
    {
        send_data1A=send_data1A+1
        timeTX1A[pkt_id] = $3
    }

    if (packet == "RX_DATA" && yo == "18" && source == "18" && dst_final == "7" &&
pkt_len == "1028" && (time < 25))
    {
        recv_data1A=recv_data1A+1
        end_RX1A = end_RX1A + $3
        timeRX1A[pkt_id] = $3
    }

    if (packet == "TX_DATA" && yo == "7" && source == "7" && dst_final == "18" && pkt_len
== "1028" && (time < 50))
    {
        send_data1B=send_data1B+1
        timeTX1B[pkt_id] = $3
    }

    if (packet == "RX_DATA" && yo == "18" && source == "18" && dst_final == "7" &&
pkt_len == "1028" && (time < 50))
    {
        recv_data1B=recv_data1B+1
        end_RX1B = end_RX1B + $3
        timeRX1B[pkt_id] = $3
    }

    if (packet == "TX_DATA" && yo == "7" && source == "7" && dst_final == "18" && pkt_len
== "1028" && (time < 75))
    {
        send_data1C=send_data1C+1
        timeTX1C[pkt_id] = $3
    }

    if (packet == "RX_DATA" && yo == "18" && source == "18" && dst_final == "7" &&

```

```

pkt_len == "1028" && (time < 75))
{
    recv_data1C=recv_data1C+1
    end_RX1C = end_RX1C + $3
    timeRX1C[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "7" && source == "7" && dst_final == "18" && pkt_len
== "1028" && (time < 100))
{
    send_data1D=send_data1D+1
    timeTX1D[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "18" && source == "18" && dst_final == "7" &&
pkt_len == "1028" && (time < 100))
{
    recv_data1D=recv_data1D+1
    end_RX1D = end_RX1D + $3
    timeRX1D[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "7" && source == "7" && dst_final == "18" && pkt_len
== "1028" && (time < 125))
{
    send_data1E=send_data1E+1
    timeTX1E[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "18" && source == "18" && dst_final == "7" &&
pkt_len == "1028" && (time < 125))
{
    recv_data1E=recv_data1E+1
    end_RX1E = end_RX1E + $3
    timeRX1E[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "7" && source == "7" && dst_final == "18" && pkt_len
== "1028" && (time < 150))
{
    send_data1F=send_data1F+1
    timeTX1F[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "18" && source == "18" && dst_final == "7" &&
pkt_len == "1028" && (time < 150))
{
    recv_data1F=recv_data1F+1
    end_RX1F = end_RX1F + $3
    timeRX1F[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "7" && source == "7" && dst_final == "18" && pkt_len
== "1028" && (time < 175))
{
    send_data1G=send_data1G+1
    timeTX1G[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "18" && source == "18" && dst_final == "7" &&
pkt_len == "1028" && (time < 175))
{
    recv_data1G=recv_data1G+1
    end_RX1G = end_RX1G + $3
    timeRX1G[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "7" && source == "7" && dst_final == "18" && pkt_len
== "1028" && (time < 200))

```

```

        {
            send_data1H=send_data1H+1
            timeTX1H[pkt_id] = $3
        }
        if (packet == "RX_DATA" && yo == "18" && source == "18" && dst_final == "7" &&
pkt_len == "1028" && (time < 200))
        {
            recv_data1H=recv_data1H+1
            end_RX1H = end_RX1H + $3
            timeRX1H[pkt_id] = $3
        }

# ----- Transmission from node 13 to 16 -----

        if (packet == "TX_DATA" && yo == "13" && source == "13" && dst_final == "16" &&
pkt_len == "1028" && (time < 25))
        {
            send_data2A=send_data2A+1
            timeTX2A[pkt_id] = $3
        }
        if (packet == "RX_DATA" && yo == "16" && source == "16" && dst_final == "13" &&
pkt_len == "1028" && (time < 25))
        {
            recv_data2A=recv_data2A+1
            end_RX2A = end_RX2A + $3
            timeRX2A[pkt_id] = $3
        }
        if (packet == "TX_DATA" && yo == "13" && source == "13" && dst_final == "16" &&
pkt_len == "1028" && (time < 50))
        {
            send_data2B=send_data2B+1
            timeTX2B[pkt_id] = $3
        }
        if (packet == "RX_DATA" && yo == "16" && source == "16" && dst_final == "13" &&
pkt_len == "1028" && (time < 50))
        {
            recv_data2B=recv_data2B+1
            end_RX2B = end_RX2B + $3
            timeRX2B[pkt_id] = $3
        }
        if (packet == "TX_DATA" && yo == "13" && source == "13" && dst_final == "16" &&
pkt_len == "1028" && (time < 75))
        {
            send_data2C=send_data2C+1
            timeTX2C[pkt_id] = $3
        }
        if (packet == "RX_DATA" && yo == "16" && source == "16" && dst_final == "13" &&
pkt_len == "1028" && (time < 75))
        {
            recv_data2C=recv_data2C+1
            end_RX2C = end_RX2C + $3
            timeRX2C[pkt_id] = $3
        }
        if (packet == "TX_DATA" && yo == "13" && source == "13" && dst_final == "16" &&
pkt_len == "1028" && (time < 100))
        {
            send_data2D=send_data2D+1
            timeTX2D[pkt_id] = $3
        }
        if (packet == "RX_DATA" && yo == "16" && source == "16" && dst_final == "13" &&

```

```

pkt_len == "1028" && (time < 100))
{
    recv_data2D=recv_data2D+1
    end_RX2D = end_RX2D + $3
    timeRX2D[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "13" && source == "13" && dst_final == "16" &&
pkt_len == "1028" && (time < 125))
{
    send_data2E=send_data2E+1
    timeTX2E[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "16" && source == "16" && dst_final == "13" &&
pkt_len == "1028" && (time < 125))
{
    recv_data2E=recv_data2E+1
    end_RX2E = end_RX2E + $3
    timeRX2E[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "13" && source == "13" && dst_final == "16" &&
pkt_len == "1028" && (time < 150))
{
    send_data2F=send_data2F+1
    timeTX2F[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "16" && source == "16" && dst_final == "13" &&
pkt_len == "1028" && (time < 150))
{
    recv_data2F=recv_data2F+1
    end_RX2F = end_RX2F + $3
    timeRX2F[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "13" && source == "13" && dst_final == "16" &&
pkt_len == "1028" && (time < 175))
{
    send_data2G=send_data2G+1
    timeTX2G[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "16" && source == "16" && dst_final == "13" &&
pkt_len == "1028" && (time < 175))
{
    recv_data2G=recv_data2G+1
    end_RX2G = end_RX2G + $3
    timeRX2G[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "13" && source == "13" && dst_final == "16" &&
pkt_len == "1028" && (time < 200))
{
    send_data2H=send_data2H+1
    timeTX2H[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "16" && source == "16" && dst_final == "13" &&
pkt_len == "1028" && (time < 200))
{
    recv_data2H=recv_data2H+1
    end_RX2H = end_RX2H + $3
    timeRX2H[pkt_id] = $3
}

# ----- Transmission from node 1 to 4 -----

```

```

    if (packet == "TX_DATA" && yo == "1" && source == "1" && dst_final == "4" && pkt_len
== "1028" && (time < 25))
    {
        send_data3A=send_data3A+1
        timeTX3A[pkt_id] = $3
    }

    if (packet == "RX_DATA" && yo == "4" && source == "4" && dst_final == "1" && pkt_len
== "1028" && (time < 25))
    {
        recv_data3A=recv_data3A+1
        end_RX3A = end_RX3A + $3
        timeRX3A[pkt_id] = $3
    }

    if (packet == "TX_DATA" && yo == "1" && source == "1" && dst_final == "4" && pkt_len
== "1028" && (time < 50))
    {
        send_data3B=send_data3B+1
        timeTX3B[pkt_id] = $3
    }

    if (packet == "RX_DATA" && yo == "4" && source == "4" && dst_final == "1" && pkt_len
== "1028" && (time < 50))
    {
        recv_data3B=recv_data3B+1
        end_RX3B = end_RX3B + $3
        timeRX3B[pkt_id] = $3
    }

    if (packet == "TX_DATA" && yo == "1" && source == "1" && dst_final == "4" && pkt_len
== "1028" && (time < 75))
    {
        send_data3C=send_data3C+1
        timeTX3C[pkt_id] = $3
    }

    if (packet == "RX_DATA" && yo == "4" && source == "4" && dst_final == "1" && pkt_len
== "1028" && (time < 75))
    {
        recv_data3C=recv_data3C+1
        end_RX3C = end_RX3C + $3
        timeRX3C[pkt_id] = $3
    }

    if (packet == "TX_DATA" && yo == "1" && source == "1" && dst_final == "4" && pkt_len
== "1028" && (time < 100))
    {
        send_data3D=send_data3D+1
        timeTX3D[pkt_id] = $3
    }

    if (packet == "RX_DATA" && yo == "4" && source == "4" && dst_final == "1" && pkt_len
== "1028" && (time < 100))
    {
        recv_data3D=recv_data3D+1
        end_RX3D = end_RX3D + $3
        timeRX3D[pkt_id] = $3
    }

    if (packet == "TX_DATA" && yo == "1" && source == "1" && dst_final == "4" && pkt_len
== "1028" && (time < 125))
    {
        send_data3E=send_data3E+1
        timeTX3E[pkt_id] = $3
    }

    if (packet == "RX_DATA" && yo == "4" && source == "4" && dst_final == "1" && pkt_len

```

```

== "1028" && (time < 125))
{
    recv_data3E=recv_data3E+1
    end_RX3E = end_RX3E + $3
    timeRX3E[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "1" && source == "1" && dst_final == "4" && pkt_len
== "1028" && (time < 150))
{
    send_data3F=send_data3F+1
    timeTX3F[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "4" && source == "4" && dst_final == "1" && pkt_len
== "1028" && (time < 150))
{
    recv_data3F=recv_data3F+1
    end_RX3F = end_RX3F + $3
    timeRX3F[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "1" && source == "1" && dst_final == "4" && pkt_len
== "1028" && (time < 175))
{
    send_data3G=send_data3G+1
    timeTX3G[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "4" && source == "4" && dst_final == "1" && pkt_len
== "1028" && (time < 175))
{
    recv_data3G=recv_data3G+1
    end_RX3G = end_RX3G + $3
    timeRX3G[pkt_id] = $3
}
if (packet == "TX_DATA" && yo == "1" && source == "1" && dst_final == "4" && pkt_len
== "1028" && (time < 200))
{
    send_data3H=send_data3H+1
    timeTX3H[pkt_id] = $3
}
if (packet == "RX_DATA" && yo == "4" && source == "4" && dst_final == "1" && pkt_len
== "1028" && (time < 200))
{
    recv_data3H=recv_data3H+1
    end_RX3H = end_RX3H + $3
    timeRX3H[pkt_id] = $3
}
}

END{

#Compare if the packet transmitted was received, through the packet ID; if it matches, it is
storing the time until to obtain the total time of sent packets that were successfully
received.

# ----- Transmission 1 -----

    for (pkt_id in timeRX1A)
    {
        if (pkt_id in timeTX1A)
        {
            start_TX1A = timeTX1A[pkt_id]

```

```

        total_start_TX1A = total_start_TX1A + start_TX1A
    }
}
for (pkt_id in timeRX1B)
{
    if (pkt_id in timeTX1B)
    {
        start_TX1B = timeTX1B[pkt_id]
        total_start_TX1B = total_start_TX1B + start_TX1B
    }
}
for (pkt_id in timeRX1C)
{
    if (pkt_id in timeTX1C)
    {
        start_TX1C = timeTX1C[pkt_id]
        total_start_TX1C = total_start_TX1C + start_TX1C
    }
}
for (pkt_id in timeRX1D)
{
    if (pkt_id in timeTX1D)
    {
        start_TX1D = timeTX1D[pkt_id]
        total_start_TX1D = total_start_TX1D + start_TX1D
    }
}
for (pkt_id in timeRX1E)
{
    if (pkt_id in timeTX1E)
    {
        start_TX1E = timeTX1E[pkt_id]
        total_start_TX1E = total_start_TX1E + start_TX1E
    }
}
for (pkt_id in timeRX1F)
{
    if (pkt_id in timeTX1F)
    {
        start_TX1F = timeTX1F[pkt_id]
        total_start_TX1F = total_start_TX1F + start_TX1F
    }
}
for (pkt_id in timeRX1G)
{
    if (pkt_id in timeTX1G)
    {
        start_TX1G = timeTX1G[pkt_id]
        total_start_TX1G = total_start_TX1G + start_TX1G
    }
}
for (pkt_id in timeRX1H)
{
    if (pkt_id in timeTX1H)
    {
        start_TX1H = timeTX1H[pkt_id]
        total_start_TX1H = total_start_TX1H + start_TX1H
    }
}

```

```

# ----- Transmission 2 -----

for (pkt_id in timeRX2A)
{
    if (pkt_id in timeTX2A)
    {
        start_TX2A = timeTX2A[pkt_id]
        total_start_TX2A = total_start_TX2A + start_TX2A
    }
}
for (pkt_id in timeRX2B)
{
    if (pkt_id in timeTX2B)
    {
        start_TX2B = timeTX2B[pkt_id]
        total_start_TX2B = total_start_TX2B + start_TX2B
    }
}
for (pkt_id in timeRX2C)
{
    if (pkt_id in timeTX2C)
    {
        start_TX2C = timeTX2C[pkt_id]
        total_start_TX2C = total_start_TX2C + start_TX2C
    }
}
for (pkt_id in timeRX2D)
{
    if (pkt_id in timeTX2D)
    {
        start_TX2D = timeTX2D[pkt_id]
        total_start_TX2D = total_start_TX2D + start_TX2D
    }
}
for (pkt_id in timeRX2E)
{
    if (pkt_id in timeTX2E)
    {
        start_TX2E = timeTX2E[pkt_id]
        total_start_TX2E = total_start_TX2E + start_TX2E
    }
}
for (pkt_id in timeRX2F)
{
    if (pkt_id in timeTX2F)
    {
        start_TX2F = timeTX2F[pkt_id]
        total_start_TX2F = total_start_TX2F + start_TX2F
    }
}
for (pkt_id in timeRX2G)
{
    if (pkt_id in timeTX2G)
    {
        start_TX2G = timeTX2G[pkt_id]
        total_start_TX2G = total_start_TX2G + start_TX2G
    }
}

for (pkt_id in timeRX2H)

```

```

        {
            if (pkt_id in timeTX2H)
            {
                start_TX2H = timeTX2H[pkt_id]
                total_start_TX2H = total_start_TX2H + start_TX2H
            }
        }

# ----- Transmission 3 -----

for (pkt_id in timeRX3A)
{
    if (pkt_id in timeTX3A)
    {
        start_TX3A = timeTX3A[pkt_id]
        total_start_TX3A = total_start_TX3A + start_TX3A
    }
}
for (pkt_id in timeRX3B)
{
    if (pkt_id in timeTX3B)
    {
        start_TX3B = timeTX3B[pkt_id]
        total_start_TX3B = total_start_TX3B + start_TX3B
    }
}
for (pkt_id in timeRX3C)
{
    if (pkt_id in timeTX3C)
    {
        start_TX3C = timeTX3C[pkt_id]
        total_start_TX3C = total_start_TX3C + start_TX3C
    }
}
for (pkt_id in timeRX3D)
{
    if (pkt_id in timeTX3D)
    {
        start_TX3D = timeTX3D[pkt_id]
        total_start_TX3D = total_start_TX3D + start_TX3D
    }
}
for (pkt_id in timeRX3E)
{
    if (pkt_id in timeTX3E)
    {
        start_TX3E = timeTX3E[pkt_id]
        total_start_TX3E = total_start_TX3E + start_TX3E
    }
}
for (pkt_id in timeRX3F)
{
    if (pkt_id in timeTX3F)
    {
        start_TX3F = timeTX3F[pkt_id]
        total_start_TX3F = total_start_TX3F + start_TX3F
    }
}

for (pkt_id in timeRX3G)

```

```

        {
            if (pkt_id in timeTX3G)
            {
                start_TX3G = timeTX3G[pkt_id]
                total_start_TX3G = total_start_TX3G + start_TX3G
            }
        }
    for (pkt_id in timeRX3H)
    {
        if (pkt_id in timeTX3H)
        {
            start_TX3H = timeTX3H[pkt_id]
            total_start_TX3H = total_start_TX3H + start_TX3H
        }
    }

    printf("\n----- TRANSMISSION 1 ----- \n") > output1
    printf("\nSEND_DATA1A: ") > output1
    printf("%i", send_data1A) > output1
    printf("\nRECV_DATA1A: ") > output1
    printf("%i", recv_data1A) > output1
    printf("\nTIME_TX1A: ") > output1
    printf("%i", total_start_TX1A) > output1
    printf("\nTIME_RX1A: ") > output1
    printf("%i", end_RX1A) > output1

    printf("\n\nSEND_DATA1B: ") > output1
    printf("%i", send_data1B) > output1
    printf("\nRECV_DATA1B: ") > output1
    printf("%i", recv_data1B) > output1
    printf("\nTIME_TX1B: ") > output1
    printf("%i", total_start_TX1B) > output1
    printf("\nTIME_RX1B: ") > output1
    printf("%i", end_RX1B) > output1

    printf("\n\nSEND_DATA1C: ") > output1
    printf("%i", send_data1C) > output1
    printf("\nRECV_DATA1C: ") > output1
    printf("%i", recv_data1C) > output1
    printf("\nTIME_TX1C: ") > output1
    printf("%i", total_start_TX1C) > output1
    printf("\nTIME_RX1C: ") > output1
    printf("%i", end_RX1C) > output1

    printf("\n\nSEND_DATA1D: ") > output1
    printf("%i", send_data1D) > output1
    printf("\nRECV_DATA1D: ") > output1
    printf("%i", recv_data1D) > output1
    printf("\nTIME_TX1D: ") > output1
    printf("%i", total_start_TX1D) > output1
    printf("\nTIME_RX1D: ") > output1
    printf("%i", end_RX1D) > output1

    printf("\n\nSEND_DATA1E: ") > output1
    printf("%i", send_data1E) > output1
    printf("\nRECV_DATA1E: ") > output1
    printf("%i", recv_data1E) > output1
    printf("\nTIME_TX1E: ") > output1
    printf("%i", total_start_TX1E) > output1
    printf("\nTIME_RX1E: ") > output1

```

```
printf("%i", end_RX1E) > output1

printf("\n\nSEND_DATA1F: ") > output1
printf("%i", send_data1F) > output1
printf("\nRECV_DATA1F: ") > output1
printf("%i", recv_data1F) > output1
printf("\nTIME_TX1F: ") > output1
printf("%i", total_start_TX1F) > output1
printf("\nTIME_RX1F: ") > output1
printf("%i", end_RX1F) > output1

printf("\n\nSEND_DATA1G: ") > output1
printf("%i", send_data1G) > output1
printf("\nRECV_DATA1G: ") > output1
printf("%i", recv_data1G) > output1
printf("\nTIME_TX1G: ") > output1
printf("%i", total_start_TX1G) > output1
printf("\nTIME_RX1G: ") > output1
printf("%i", end_RX1G) > output1

printf("\n\nSEND_DATA1H: ") > output1
printf("%i", send_data1H) > output1
printf("\nRECV_DATA1H: ") > output1
printf("%i", recv_data1H) > output1
printf("\nTIME_TX1H: ") > output1
printf("%i", total_start_TX1H) > output1
printf("\nTIME_RX1H: ") > output1
printf("%i", end_RX1H) > output1

printf("\n----- TRANSMISSION 2 ----- \n") > output2
printf("\nSEND_DATA2A: ") > output2
printf("%i", send_data2A) > output2
printf("\nRECV_DATA2A: ") > output2
printf("%i", recv_data2A) > output2
printf("\nTIME_TX2A: ") > output2
printf("%i", total_start_TX2A) > output2
printf("\nTIME_RX2A: ") > output2
printf("%i", end_RX2A) > output2

printf("\n\nSEND_DATA2B: ") > output2
printf("%i", send_data2B) > output2
printf("\nRECV_DATA2B: ") > output2
printf("%i", recv_data2B) > output2
printf("\nTIME_TX2B: ") > output2
printf("%i", total_start_TX2B) > output2
printf("\nTIME_RX2B: ") > output2
printf("%i", end_RX2B) > output2

printf("\n\nSEND_DATA2C: ") > output2
printf("%i", send_data2C) > output2
printf("\nRECV_DATA2C: ") > output2
printf("%i", recv_data2C) > output2
printf("\nTIME_TX2C: ") > output2
printf("%i", total_start_TX2C) > output2
printf("\nTIME_RX2C: ") > output2
printf("%i", end_RX2C) > output2

printf("\n\nSEND_DATA2D: ") > output2
printf("%i", send_data2D) > output2
printf("\nRECV_DATA2D: ") > output2
```

```

printf("%i", recv_data2D) > output2
printf("\nTIME_TX2D: ") > output2
printf("%i", total_start_TX2D) > output2
printf("\nTIME_RX2D: ") > output2
printf("%i", end_RX2D) > output2

printf("\n\nSEND_DATA2E: ") > output2
printf("%i", send_data2E) > output2
printf("\nRECV_DATA2E: ") > output2
printf("%i", recv_data2E) > output2
printf("\nTIME_TX2E: ") > output2
printf("%i", total_start_TX2E) > output2
printf("\nTIME_RX2E: ") > output2
printf("%i", end_RX2E) > output2

printf("\n\nSEND_DATA2F: ") > output2
printf("%i", send_data2F) > output2
printf("\nRECV_DATA2F: ") > output2
printf("%i", recv_data2F) > output2
printf("\nTIME_TX2F: ") > output2
printf("%i", total_start_TX2F) > output2
printf("\nTIME_RX2F: ") > output2
printf("%i", end_RX2F) > output2

printf("\n\nSEND_DATA2G: ") > output2
printf("%i", send_data2G) > output2
printf("\nRECV_DATA2G: ") > output2
printf("%i", recv_data2G) > output2
printf("\nTIME_TX2G: ") > output2
printf("%i", total_start_TX2G) > output2
printf("\nTIME_RX2G: ") > output2
printf("%i", end_RX2G) > output2

printf("\n\nSEND_DATA2H: ") > output2
printf("%i", send_data2H) > output2
printf("\nRECV_DATA2H: ") > output2
printf("%i", recv_data2H) > output2
printf("\nTIME_TX2H: ") > output2
printf("%i", total_start_TX2H) > output2
printf("\nTIME_RX2H: ") > output2
printf("%i", end_RX2H) > output2

printf("\n----- TRANSMISSION 3 ----- \n") > output3
printf("\nSEND_DATA3A: ") > output3
printf("%i", send_data3A) > output3
printf("\nRECV_DATA3A: ") > output3
printf("%i", recv_data3A) > output3
printf("\nTIME_TX3A: ") > output3
printf("%i", total_start_TX3A) > output3
printf("\nTIME_RX3A: ") > output3
printf("%i", end_RX3A) > output3

printf("\n\nSEND_DATA3B: ") > output3
printf("%i", send_data3B) > output3
printf("\nRECV_DATA3B: ") > output3
printf("%i", recv_data3B) > output3
printf("\nTIME_TX3B: ") > output3
printf("%i", total_start_TX3B) > output3
printf("\nTIME_RX3B: ") > output3
printf("%i", end_RX3B) > output3

```

```
printf("\n\nSEND_DATA3C: ") > output3
printf("%i", send_data3C) > output3
printf("\nRECV_DATA3C: ") > output3
printf("%i", recv_data3C) > output3
printf("\nTIME_TX3C: ") > output3
printf("%i", total_start_TX3C) > output3
printf("\nTIME_RX3C: ") > output3
printf("%i", end_RX3C) > output3

printf("\n\nSEND_DATA3D: ") > output3
printf("%i", send_data3D) > output3
printf("\nRECV_DATA3D: ") > output3
printf("%i", recv_data3D) > output3
printf("\nTIME_TX3D: ") > output3
printf("%i", total_start_TX3D) > output3
printf("\nTIME_RX3D: ") > output3
printf("%i", end_RX3D) > output3

printf("\n\nSEND_DATA3E: ") > output3
printf("%i", send_data3E) > output3
printf("\nRECV_DATA3E: ") > output3
printf("%i", recv_data3E) > output3
printf("\nTIME_TX3E: ") > output3
printf("%i", total_start_TX3E) > output3
printf("\nTIME_RX3E: ") > output3
printf("%i", end_RX3E) > output3

printf("\n\nSEND_DATA3F: ") > output3
printf("%i", send_data3F) > output3
printf("\nRECV_DATA3F: ") > output3
printf("%i", recv_data3F) > output3
printf("\nTIME_TX3F: ") > output3
printf("%i", total_start_TX3F) > output3
printf("\nTIME_RX3F: ") > output3
printf("%i", end_RX3F) > output3

printf("\n\nSEND_DATA3G: ") > output3
printf("%i", send_data3G) > output3
printf("\nRECV_DATA3G: ") > output3
printf("%i", recv_data3G) > output3
printf("\nTIME_TX3G: ") > output3
printf("%i", total_start_TX3G) > output3
printf("\nTIME_RX3G: ") > output3
printf("%i", end_RX3G) > output3

printf("\n\nSEND_DATA3H: ") > output3
printf("%i", send_data3H) > output3
printf("\nRECV_DATA3H: ") > output3
printf("%i", recv_data3H) > output3
printf("\nTIME_TX3H: ") > output3
printf("%i", total_start_TX3H) > output3
printf("\nTIME_RX3H: ") > output3
printf("%i", end_RX3H) > output3

close(output1)
close(output2)
close(output3)
}
```

REFERENCES

- [Ali 2010] Saleh Ali K. Al-Omari, Putra Sumari. "An Overview of Mobile Ad hoc Networks for the Existing Protocols and Applications". International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks (Graph-Hoc), Vol.2, No.1, Pages 87-110. March 2010.
- [AWK 2011] The GNU Awk User's Guide. This is Edition 4 of *GAWK: Effective AWK Programming: A User's Guide for GNU Awk*. For the 4.0.0 (or later) version of the GNU implementation of AWK. 2011.
<http://www.gnu.org/software/gawk/manual/gawk.html>
- [Benzaid 2002] Mounir Benzaid, Pascale Minet, Khaldoun Al Agha. "Integrating Fast Mobility in the OLSR Routing Protocol". Proceedings of IEEE Conference on Mobile and Wireless Communications Networks. No. 4510. ISSN: 0249. ISRN INRIA/RR-4510-FR+ENG. June, 2002.
- [Bianchi 2000] Giuseppe Bianchi. "Performance Analysis of the IEEE 802.11 Distributed Coordination Function". IEEE Journal on Selected Areas in Communications. 2000. Volume 18, Issue 3. Pages 535-547. ISSN: 0733-8716. DOI: 10.1109/49.840210.
- [Brahmi 2009] N. Brahmi, M. Boussedjra, J. Mouzna, M. Bayart. "Adaptive Movement Aware Routing for Vehicular Ad hoc Networks". Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly. IWCMC '09. ACM. New York, NY, USA. 2009. Pages 1310-1315. ISBN: 978-1-60558-569-7. DOI: 10.1145/1582379.1582666.
- [Campos 2011] Xavier Campos Ríos, Diego Pastor Pérez. Projecte Final de Carrera. Universidad Politècnica de Catalunya. "Performance Evaluation of Realistic Scenarios for Vehicular Ad hoc Networks with Citymob and NCTUns".
- [CAR-2-CAR] CAR-2-CAR Communication Consortium. <http://www.car-2-car.org/>
- [CARLINK] CELTIC Telecommunication Solutions. CARLINK: Wireless Traffic Service Platform for Linking Cars. Project Consortium. <http://carlink.lcc.uma.es/>
- [Chaudet 2001] Claude Chaudet, Isabelle Guérin Lassous. "BRuIT: Bandwidth Reservation under Interference influence". INRIA. No. 4308. 2001. ISSN: 0249-6399
- [Citymob 2008] Francisco J. Martinez, Juan Carlos Cano, Carlos T. Calafate, Pietro Manzoni. "Citymob: A Mobility Model Pattern Generator for VANETs". IEEE International Conference on Communications Workshops. ICC Workshops 2008. Pages 370-374. ISBN: 978-1-4244-2052-0. DOI: 10.1109/ICCW.2008.76
- [Citymob 2009] Universidad Politècnica de Valencia. Grupo de Redes de Computadoras (GRC). 2009
<http://www.grc.upv.es/Software/citymob.html>
- [CVIS] CVIS. Cooperative Vehicle-Infrastructure Systems. European Commission Information Society and Media. <http://www.cvisproject.org/>
- [DSRC] ITS Standard Advisory. Department of Transportation. Dedicated Short-Range Communication (DSRC). http://www.standards.its.dot.gov/Documents/advisories/dsrc_advisory.htm
- [Eenennaam 2008] E.M. Van Eenennaam. "A Survey of Propagation Models used in Vehicular Ad hoc Network (VANET)". Paper written for course Mobile Radio Communication, University of Twente. June 2008.
- [EstiNet 2010] EstiNet. Technologies Real Applications, Protocols & Performance. <http://www.estinet.com/>
- [Fedora 2009a] Fedora. Get Fedora. Freemod. Friend. Features. First. A Red Hat-Sponsored Community Project. <http://archive.fedoraproject.org/pub/archive/fedora/linux/releases/12/Fedora/i386/iso/Fedora-12-i386-DVD.iso>
- [Fedora 2009b] Fedora 12. Documentation. Installation Guide Installing Fedora 12 on x86 AMD64, and Intel 64 architectures. Edition 1.0. Fedora Documentation Project.
http://docs.fedoraproject.org/en-US/Fedora/12/html/Installation_Guide/
- [Garcia 2007] Carlos F. Garcia Hernández, Pablo H Ibarguéengoytia González, Joaquín García Hernandez, Jesús A. Pérez Díaz. "Wireless Sensor Networks and Applications: a Survey." International Journal of Computer Science and Network Security, VOL. 7 No.3. Pages 264-273. 2007.
- [Gerla 2005] Ad hoc networks, Technologies and Protocols. Edited by Prasant Mohapatra and Srikanth Krishnamurthy. By Mario Gerla. Chapter 1, "Ad hoc networks Emerging Applications, Design Challenges and Future Opportunities". Pages 1-22. ISBN: 0-387-22689-3. eBook ISBN: 0-387-22690-7. 2005 Springer Science + Business Media, Inc.

-
- [Gong 2007] Jiayu Gong, Cheng-Zhong Xu, James Holle. “*Predictive Directional Greedy Routing in Vehicular Ad hoc Networks*”. 27th International Conference on Distributed Computing Systems Workshops. ICDCSW '07. ISSN: 1545-0678. DOI: 10.1109/ICDCSW.2007.65.
- [Han 2004] Lu Han. “*Wireless Ad hoc Networks*”. October 8, 2004.
- [IEEE 802.11 WG] QUICK GUIDE TO IEEE 802.11 WG & ACTIVITIES. Types of IEEE 802.11 activities. http://www.ieee802.org/11/QuickGuide_IEEE_802_WG_and_Activities.htm
- [IEEE 802.11] IEEE Standard 802.11e-2005. “*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*”. <http://standards.ieee.org/findstds/standard/802.11-2007.html>
- [IEEE 802.11e] IEEE Standard 802.11e-2005. “*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment: Medium Access Method (MAC) Quality of Service Enhancements*”. <http://standards.ieee.org/findstds/standard/802.11e-2005.html>
- [IEEE 802.11p] IEEE Standard 802.11p-2010. “*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 6: Wireless Access in Vehicular Environments*”. <http://standards.ieee.org/findstds/standard/802.11p-2010.html>
- [IEEE P1609] IEEE 1609 – Family of Standards for Wireless Access in Vehicular Environments (WAVE) http://www.standards.its.dot.gov/fact_sheet.asp?f=80
- [ITS America] “*Intelligent Transportation Systems*”. Joint Program Office. U.S. Department of Transportation. Research and Innovative Technology Administration (RITA). <http://www.its.dot.gov/>
- [ITS Spain] “*Intelligent Transportation Systems*”. Spain. <http://www.itsspain.com/itsspain/>
- [ITU E.800] International Telecommunication Union. E.800: Definitions of terms related to quality of service. Recommendation E.800 (09/08). <http://www.itu.int/rec/T-REC-E.800-200809-I>
- [Johnson 1996] David B. Johnson, David A. Maltz. “*Dynamic Source Routing in Ad hoc Wireless Networks*”. In *Mobile Computing*, Vol, 353. 1996
- [Karp 2000] Brad Karp, H.T. Kung. “*GPSR: Greedy Perimeter Stateless Routing for Wireless Networks*”. Proceeding of the 6th Annual International Conference on Mobile Computing and Networking. MobiCom'00. ACM. New York, NY, USA. 2000. Pages 243-254. ISBN: 1-58113-197-6. DOI: 10.1145/345910.345953
- [Lochert 2003] Christian Lochert, Hannes Hartenstein, Jing Tian, Holger Fuessler, Dagmar Hermann, Martin Mauve. “*A Routing Strategy for Vehicular Ad hoc Networks in City Environments*”. Proceedings of IEEE Intelligent Vehicles Symposium. Pages 156-161. ISBN: 0-7803-7848-2. DOI: 10.1109/IVS.2003.1212091. June, 2003.
- [Lochert 2005] Christian Lochert, Martin Mauve, Holger Fussler, Hannes Hartenstein. “*Geographic Routing in City Scenarios*”. ACM SIGMOBILE Mobile Computing and Communications Review. Volume 9, Issue 1. New York, NY, USA. DOI: 10.1145/1055959.1055970. Springer Link. January 2005.
- [MANET WG] IETF datatracker.ietf.org Mobile Ad-hoc Networks (manet). Description of Working Group. <http://datatracker.ietf.org/wg/manet/charter/>
- [Moustafa 2008] Hassnaa Moustafa, Sidi Mohammed Senouci, Moez Jerbi. Vehicular Networks: Techniques, Standards and, Applications. *Chapter 1 “Introduction to Vehicular Networks”*. Pages 1-20. AU857'C001. 2008.
- [NCTUns 2010] SIMREAL TECHNOLOGY. N.S.L. Software that makes simulations seem real. Shie-Yuan Wang, Chih-Liang Chou, Chih-Che Lin and Chih-Hua Huang. “*The Protocol Developers Manual for NCTUns 6.0 Network Simulator*”. Produced and maintained by Network and System Laboratory, Department of Computer Science, National Chiao Tung University, Taiwan. Last update: January 15, 2010.
- [NCTUns] SIMREAL TECHNOLOGY. N.S.L. Software that makes simulations seem real. Shie-Yuan Wang, Chih-Liang Chou, Chih-Che Lin. “*The GUI User Manual for the NCTUns 6.0 Network Simulator and Emulator*”. Produced and maintained by Network and System Laboratory, Department of Computer Science, National Chiao Tung University, Taiwan. Last update: September 1, 2009.
- [NOW] NOW. Network on Wheels. Federal Ministry of Education and Research. <http://www.network-on-wheels.de/>
- [Nzouonta 2009] Josiane Nzouonta, Neeraj Rajgure, Guiling Wang, Cristian Borcea. “*VANET Routing on City Roads Using Real-Time Vehicular Traffic Information*”. IEEE Transactions on Vehicular Technology. IEEE Vehicular Technology Society. 2009. Volume 58, Issue 7. Pages 3609-3626. ISSN: 0018-9545. DOI: 10.1109/TVT.2009.2014455
- [Perkins 1994] Charles E. Perkins, Pravin Bhagwat. “*Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers*”. Proceedings of the Conference on Communications Architectures, Protocols and Applications. SIGCOMM '94. ISBN: 0-89791-682-4. DOI: 10.1145/190314.190336.
-

-
- Newslettter, ACM, SIGCOMM, Computer Communication Review. ACM. New York, NY, USA. 1994. Volume 24, Issue 4. DOI: 10.1145/190809.190336.
- [Perkins 1999] Charles E. Perkins, Elizabeth M. Royer. *"Ad-hoc On-Demand Distance Vector Routing"*. Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications. WMSCA'99. Pages 90-100. ISBN: 0-7695-0025-0. DOI: 10.1109/MCSA.1999.749281.
- [PRECIOSA] PRECIOSA. Privacy Enabled Capability In Co-operative Systems and Safety Applications. Safety. Information Society and Media. <http://www.preciosa-project.org/>
- [PreVENT] PreVENT. European Commission Information Society and Media. WILLWARN a PreVENT Project. http://www.preventip.org/en/prevent_subprojects/safe_speed_and_safe_following/willwarn/willwarn_02.htm
- [Ramanathan 2002] Ram Ramanathan, Jason Redi. BBN Technologies. *"A Brief Overview of Ad hoc Networks: Challenges and Directions"*. IEEE Communications Magazine, 50th Anniversary Commemorative Issue. Pages 20-22. May 2002.
- [Rennesse 2004] Ronan de Renesse, Mona Ghassemian, Vasilis Friderikos, A. Hamid Aghvami. *"QoS Enabled Routin in Mobile Ad ho Networks"*. 5th IEEE Conference on 3G Mobile Communication Technologies. 3G 2004. Pages 678-682. ISSN: 0537-9989. ISBN: 0-86341-388-9. DOI: 10.1049/cp:20040762
- [Rennesse 2007] R. de Renesse, V. Friderikos, H. Aghvami. *"Cross-layer Cooperation for Accurate Admission Control Decisions in Mobile Ad hoc Networks"*. Institution of Engineering and Technology Communications, IET. 2007. Volume 1, Issue 4. Pages 577-586. ISSN: 1751-8628. DOI: 10.1049/iet-com:20060243.
- [RFC 3561] Internet Engineering Task Force. Network Working Group, RFC. *"Ad hoc On-demand Distance-Vector"*. <http://tools.ietf.org/pdf/rfc3561.pdf>
- [Rontala 2011] Prabhakar Rontala Subramaniam, Arunkumar Thangavelu, Chitra Venugopal. *"QoS for Highly Dynamic Vehicular Ad hoc Network Optimally"*. 11th International Conference on ITS Telecommunications. ITST '11. Pages 405-411. ISBN: 978-1-61284-668-2. DOI: 10.1109/ITST.2011.6060091
- [SAFESPOT] SAFESPOT. Cooperative vehicles and road infrastructure for road safety. European Commission Information Society and Media. <http://www.safespot-eu.org/>
- [Salem 2006] Osman Salem, Abdelmalek Benzekri. *"Towards Absolute End-to-End QoS in Ad hoc Networks"*. Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services. AICT/ICIW 2006. ISBN: 0-7695-2522-9. DOI: 10.1109/AICT-ICIW.2006.195
- [Sarr 2008] Cheikh Sarr, Claude Chaudet, Guillaume Chelius, Isabelle Guerien Lassous. *"Bandwidth Estimation for IEEE 802.11-Based Ad hoc Networks"*. IEEE Computer Society. IEEE Transactions on Mobile Computing. Vol. 7, No. 10. Pages 1228-1241. ISSN: 1536-1233. October 2008. DOI: 10.1109/TMC.2008.41
- [Seet 2004] Boon-chong Seet, Genping Liu, Bu-sung Lee, Chuand-heng Foh, Keok-kee Lee. *"A-STAR: A Mobile Ad hoc Routing Strategy for Metropolis Vehicular Communications"*. Networking Technologies Services and Protocols Performance of Computer and Communication Networks Mobile and Wireless Communication. NETWORKING 2004. Springer. DOI: 10.1007/978-3-540-24693-0_81.
- [SEVECOM] SEVECOM. Secure Vehicle Communication. <http://www.sevecom.org>
- [Strauss 2003] Jacob Strauss, Dina Katabi, Frans Kaashoek. *"A Measurement Study of Available Bandwidth Estimation Tools"*. Proceeding of the 3rd ACM SIGCOMM Conference on Internet Measurement. ACM. New York, NY, USA. 2003. ISBN: 1-58113-773-7. DOI: 10.1145/948205.948211
- [Taneja 2007] Kavita Taneja, R.B. Patel. *"An Overview of Mobile Ad hoc Networks: Challenges and Future"*. Proceedings of National Conference on Challenges & Opportunities in Information Technology (COIT-2007) RIMT-IET, Mandi Gobindgarh. Pages 133-135. March 23, 2007.
- [Tee 2010] C. A. T. H. Tee, A. Lee. *"A Novel Routing Protocol – Junction Based Adaptive Reactive Routing (JARR) for VANET in City Environments"*. European Wireless Conference. 2010. Pages 1-6. ISBN: 978-1-4244-5999-5. DOI: 10.1109/EW-2010.5483386.
- [Tian 2003] Jing Tian, Lu Han, Kurt Rothermel. *"Spatially Aware Packet Routing for Mobile Ad hoc Inter-Vehicle Radio Networks"*. Proceedings of IEEE Intelligent Transportation Systems. 2003. Vol 2. Pages 1546-1551. ISBN: 0-7803-8125-4. DOI: 10.1109/ITSC.2003.1252743.
- [Tripp] Carolina Tripp Barba, Ahmad Mohamed Mezher, Mónica Aguilar Igartua, Isabelle Guerin Lassaus Cheikh Sarr. *Available Bandwidth-aware Routing in Urban Vehicular Ad-hoc Networks*. (Not yet published).
-

-
- [Van Nam 2011] Nguyen Van Nam, Isabelle Guerin Lassous, Victor Moraru, Cheik Sarr. *"Retransmissions-based Available Bandwidth Estimation in IEEE 802.11-based Multihop Wireless Networks"*. Proceedings of 14th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems. MSWiM'11. ACM. New York, NY, USA. 2011. ISBN:978-1-4503-0898-4. DOI: 10.1145/2068897.2068961.
- [VMware 2009] VMware Fusion. Getting Started with VMware Fusion. VMware Fusion for Mac OS X. Item: EN-000189-00. http://www.vmware.com/pdf/fusion_getting_started_30.pdf
- [VMware 2011] VMware. Download VMware Fusion. http://downloads.vmware.com/d/info/desktop_end_user_computing/vmware_fusion/3_0
- [Wei Lin 2010] Yun Wei Lin, Yun Shyan Chen, Sing Ling Lee. *"Routing Protocols in Vehicular Ad hoc Networks: A Survey and Future Perspectives"*. Journal of Information Science and Engineering 26, Pages 913-932. 2010.
- [Wu 2003] Dapeng Wu, Rohit Negi. *"Effective Capacity: A Wireless Link Model for Support Quality of Service"*. IEEE Transactions on Wireless Communications. 2003. Volume 2, Issue 4. Pages 630-643. ISSN: 1536-1276. DOI: 10.1109/TWC.2003.814353
- [Wu 2007] Haitao Wu, Xin Wang, Yunxin Liu, Qian Zhang, Zhi-Li Zhang. *"SoftMAC: Layer 2.5 MAC for VoIP Support in Multi-hop Wireless Networks"*. IEEE Transactions on Mobile Computing. 2007. Volume 6, Issue 1. DOI: 10.1109/TCM.2007.15
- [Xu 2003] Kaixin Xu, Ken Tang, Rajive Bagrodia, Mario Gerla, Michael Bereschinsky. *"Adaptive Bandwidth Management and QoS Provisioning in Large Scale Ad hoc Networks"*. Proceeding of the IEEE Conference on Military – Volume II. MILCOM '03. IEEE Computer Society Washington, DC, USA. 2003. ISBN: 0-7803-8140-8
- [Yahya 2011] Mobile Ad-Hoc Networks: Applications, Edited by Xin Wang. By Zaydoun Yahya Rawashedeh and Syed Masud Mahmud. *Chapter 2: Communications in Vehicular Networks*. Published by InTech. 2011. ISBN: 978-953-307-416-0
- [Yan 2010] Gongjun Yan, Danda B. Rawat, Bhed B. Bista. *"Provisioning Vehicular Ad hoc Networks with Quality of Service"*. 2010 International Conference on Broadband, Wireless Computing, Communication and Applications. BWCCA 2010. Pages 102-107. ISBN: 978-1-4244-8448-5. DOI: 10.1109/BWCCA.2010.55.
- [Yang 2005] Yaling Yang, Robin Kravets. *"Contention-aware Admission Control for Ad hoc Networks"*. IEEE Transactions on Mobile Computing. 2005. Volume 4, Issue 4. Pages 363-377. ISSN: 1536-1233. DOI: 10.1109/TMC.2005.52.
- [Yang 2008] Qing Yang, Alvin Lim, Prathima Agrawal. *"Connectivity Aware Routing (CAR) in Vehicular Networks"*. IEEE Wireless Communications and Networking Conference. WCNC 2008. Pages 2218-2223. ISBN: 978-1-4244-1997-5. DOI: 10.1109/WCNC.2008.392
- [Zhai 2004] Hongqiang Zhai, Xiang Chen, Yuguang Fang. *"A Call Admission Rate Control Scheme for Multimedia Support over IEEE 802.11 Wireless LANs"*. First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks. QSHINE 2004. Pages 76-83. ISBN: 0-7695-2233-5. DOI: 10.1109/QSHINE.2004.1
- [Zhao 2009] Haitao Zhao, Emiliano Garcia Palacios Jibo Wei, Yong Xi. *"Accurate Available Bandwidth Estimation in IEEE.802.11-based Ad hoc Networks"*. Published in Journal Computer Communications. 2009. Volume 32, Issue 6. Pages 1050-1057. Elsevier Science Publishers B. V. Amsterdam. The Netherlands. DOI: 10.1016/j.comcom.2008.12.03.
- [Zhaomin 2006] Mo Zhaomin, Zhu Hao, Makki Kia, Pissinou Niki. *"MURU: A Multihop-Hop Routing Protocol for Urban Vehicular Ad hoc Networks"*. Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services. Pages 1-8. ISBN: 1-4244-0498-3. 10.1109/MMOBQ.2006.340406. 2006.
-