

Data dissemination in vehicular networks



Gianluca Grilli

Department of Computer Science, Systems and Production

University of Rome "Tor Vergata"

*Philosophiæ Doctor (PhD) dissertation in
Computer Science and Automation Engineering
(cycle XXI)*

Advisor and Coordinator: Prof. Daniel P. Bovet

16 June, 2010

PhD Board of Examiners (in alphabetical order):

Prof. Bruno Cicciani

Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza"
via Ariosto 25, 00185 Roma

Prof. Vittorio Cortellessa

Dipartimento di Informatica, Università dell'Aquila
via Vettoio, Coppito 67100 L'Aquila

Prof. Paola Velardi

Dipartimento di Informatica, Università di Roma "La Sapienza"
via Salaria 113, 00198 Roma

Day of the defense: 16 June, 2010

Signature from head of PhD committee:

Abstract

Data dissemination consists on spreading a large amount of information to all nodes belonging to a network and its peculiar characteristics make this goal particularly interesting and challenging. However, developing efficient data dissemination schemes for vehicular ad-hoc networks (VANETs) is still an open issue due to the broadcast nature of the channel and to the need of managing all data transmissions in a distributed way. The former leads to a lot of problems related to the channel contention, collisions and interference. The latter requires to define algorithms which exploit only local information of the network and which are scalable and robust to the node mobility.

The focus of this thesis is to investigate the data dissemination process in VANETs by defining and developing a new communication protocol (CORP), efficient in terms of limited computational complexity, low latency, high speed of dissemination, high delivery ratio, better usage of the wireless channel and, consequently, better energy consumption. The basic idea relies on the application of the fountain codes theory to vehicular networks in order to implement a true I2V2V (*infrastructure-to-vehicle-to-vehicle*) communication paradigm.

Results gathered from extensive simulation show the feasibility of such a solution, which could be implemented in real on-board communication devices for *Intelligent Transportation Systems* (ITS).

*"There are three kinds of death in this world. There's heart death, there's
brain death, and there's being off the network."*

- Guy Almes

*To my loved grandparents,
your memories will always belong to my heart.*

Acknowledgements

I would first and foremost like to thank my advisor, Prof. Daniel Bovet, without whom my PhD experience would not have been possible. His dedication and passion, together with his excellent analytical skills are great assets to those who are fortunate enough to work with him.

Many thanks to Prof. Mario Gerla, my advisor at the Network Research Lab (NRL) of the University of California Los Angeles (UCLA). I will never forget him as well as the pleasant (but fruitful) weekly meetings held at NRL. I would also like to thank the whole PhD group at P1B-03 office for their inputs and cooperation during the last years and, above all, for their friendship. In this respect, a special thank goes to Fabio Dellutri, Emanuele Galli and Emiliano Betti. Andrea Bastoni also deserves a great deal of credit especially for his invaluable logistic support and assistance during my "last" operating system course.

I cannot omit my friends from the USA experience, namely, Ali Hamidian, Paolo Di Rico, Paolo Lutterotti, Eugenio Giordano, Ilias Leontiadis, Pasquale Cataldi, Alexandro Sentinelli, Andrea Tomatis, Alessandro Bisacco and Giovanni Mazzeo.

Finally, I wish to extend my deepest gratitude to my family and my sweet love, "Patata", who both have never wavered in their support of me in all of my ventures in life.

The last word of gratitude goes to Chopin, my lovely labrador, for its countless love.

Contents

List of Figures	vii
List of Tables	xv
1 Introduction	1
1.1 Layout of the Thesis	3
2 Preliminary concepts	5
2.1 Mobile Ad-hoc NETwork (MANET)	5
2.2 Vehicular Ad-hoc NETwork (VANET)	7
2.3 Network requirements in VANETs applications	10
2.3.1 Mobility	10
2.3.2 Permanent access	11
2.3.3 Location Awareness	12
2.3.4 Time Awareness	12
2.3.5 Penetration rate dependency	13
2.3.6 Geocast capability	13
2.4 VANETs' common communication schemes	14
2.5 Communication technology in VANETs	16
2.5.1 WLAN - IEEE 802.11	17
2.5.2 WiMAX	24
2.5.3 Bluetooth	25
2.5.3.1 Bluetooth Architecture	26
2.5.4 ZigBee	27
2.5.5 DSRC and 802.11p (or "WAVE")	28
2.5.6 Frequency and channel allocation	32

CONTENTS

2.5.7	Cellular networks	35
2.5.8	Radio Data System and Traffic Message Channel	36
2.5.9	Satellite	37
2.6	Overview of Communication systems in VANETs	37
2.7	The importance of Geocast protocols in VANETs	40
3	Mobility models in vehicular networks	43
3.1	MANET'S MOBILITY MODELS	44
3.1.1	Entity models	44
3.1.2	Group models	49
3.2	VANET'S MOBILITY MODELS	50
3.2.1	Factors affecting mobility in VANETs	53
3.2.2	Freeway	54
3.2.3	Manhattan	55
3.2.4	City Section Mobility	55
3.2.5	Rice University Model (RUM)	57
3.2.6	Stop Sign Model (SSM)	57
3.2.7	Traffic Sign Model (TSM)	58
3.2.8	STRAW	58
3.2.9	Gorgorin et al. model	59
3.2.10	Metrics for mobility models	61
4	Vehicular Network Simulators	67
4.1	Introduction to VANET simulation	68
4.2	Mobility simulators	68
4.2.1	TSIS-CORSIM	69
4.2.2	VisSim	70
4.2.3	PARAMICS	70
4.2.4	Canumobisim and Vanetmobisim	73
4.2.4.1	Microbility features in Vanetmobisim	77
4.2.5	SUMO and MOVE	80
4.3	Network simulators	81
4.3.1	OPNET	82
4.3.2	GloMoSim and QualNET	83

4.3.2.1	Discrete event simulation	86
4.3.2.2	Qualnet protocol stack	86
4.3.3	NS-2	90
4.3.4	J-Sim	91
4.3.5	OMNeT++	91
4.3.6	SWANS	91
4.4	Tightly Integrated Simulators	92
4.4.1	SWANS++	94
4.4.2	GrooveNeT	95
4.4.3	TraNS	96
4.4.4	Veins	98
4.4.5	NCTUns	98
4.4.6	ASH	99
4.5	Scalability of VANET simulators	102
5	Data dissemination survey	107
5.1	Types of information involved in the dissemination process	107
5.2	Data dissemination techniques	114
5.3	Information sharing via Network Coding technique	115
5.3.1	Introduction to Network Coding	115
5.3.2	What is Network Coding	117
5.3.3	Linear Network Coding	120
5.3.4	The coding and decoding process	122
5.3.5	Benefits of Network Coding	124
5.4	Example of Content distribution applications for VANETs	127
5.4.1	SPAWN (CarTorrent)	127
5.4.2	CodeTorrent	130
6	Data dissemination with rateless codes	135
6.1	Before you start: digital fountain codes	135
6.1.1	LT codes	136
6.1.2	Tornado codes	140
6.1.3	Raptor codes	141
6.2	An application of fountain codes: CORP	144

CONTENTS

6.2.1	Introduction	144
6.2.2	Related work	146
6.2.3	Proposed approach	148
6.2.4	Packet definition	153
6.2.5	Communications	154
6.2.5.1	Phase 1: AP to MN	154
6.2.5.2	Phase 2: MN to MN	155
6.2.6	CORP-TCP	157
6.2.7	Simulations and results	159
7	Conclusions	171
	References	173

List of Figures

2.1	Application requirements in VANETs. Please, note that no asterisk means "none or not needed" requirement, "*" means "needed", "**" stands for "suited" requirement.	11
2.2	Geocasting forwarding schemes. Typical geocasting forwarding schemes: <i>unicast</i> (A), <i>broadcast</i> (B) and <i>topologically scoped broadcast</i> (C).	14
2.3	InV communications. The InV communication scheme is mainly used to create a specialized sensor networks within the vehicle. It is very useful when you want to control vehicle's on board devices and, at the same time, act on the motion behaviour in response to a received emergency message. (Mercedes Benz, SL 500 - source: Daimler Benz)	15
2.4	Communication schemes in VANETs. The figure shows the communication schemes typically used in a VANET: (a) Vehicle-to-Vehicle (V2V), (b) Vehicle-to-Infrastructure (V2I) and Infrastructure-to-Vehicle (I2V).	16
2.5	Overview of the application of network technologies used in vehicle ad-hoc networks. The figure shows a vehicular networking scheme where the most important technologies at level-three are included in an integral communication solution. Finally, an overlay architecture using cellular networks shows the feasibility of this technology to enable vehicular communications.	17
2.6	Wireless technologies for InV communications.	18
2.7	802.11a channels.	20
2.8	802.11b/g channels.	21
2.9	Example of a bluetooth scatternet.	27

LIST OF FIGURES

2.10	DSRC Standards DSRC standards and communication stack.	30
2.11	Wireless Technologies for V2V and V2I communications. With the rapid development of information technologies, there are a number of wireless technologies which are potential for wireless InV, V2V and V2I communications.	32
2.12	Frequency and channel allocation. 30 MHz frequency band dedicated for safety-related car-to-car and car-to-infrastructure communication where SCH1 is the first service channel, SCH2 is the secondary service channel, and CCH is the control channel.	34
2.13	CAR-2-X communication scheme. Typical VANET scenario showing a CAR-2-X communication system and involved protocols of the IEEE 802 family. As shown in the figure, the CAR-2-X communication system consists of three domains: the <i>in-vehicle domain</i> , the <i>ad hoc domain</i> , and the <i>infrastructure domain</i>	39
2.14	Geocasting forwarding schemes. Typical geocasting forwarding schemes: <i>unicast</i> (A), <i>broadcast</i> (B) and <i>topologically scoped broadcast</i> (C).	41
3.1	Random Walk Mobility model. Traveling pattern of a MN using a probabilistic version of RWM.	46
3.2	Random-Wapoint model In RWP, mobile nodes choose randomly a destination and move towards it with a constant velocity.	47
3.3	Random-Wapoint trace. Typical RWP's motion trace in a 500x500 area.	47
3.4	Random-Waypoint neighbors percentage. Average neighbors percentage vs. time.	48
3.5	RPGM model Movements of three nodes according to the RPGM mobility model.	51
3.6	RPGM group pattern. Traveling group pattern of three nodes using the RPGM mobility model.	51
3.7	RPGM group pattern (5 mobile nodes). Traveling group pattern of five nodes using the RPGM mobility model.	52
3.8	City Section Mobility Model. Traveling pattern of a MN using the CSMM.	56

3.9	Psycho-physical vehicular interaction. The Δx axis divides positive (vehicle approaching front car) and negative (front vehicle increasing its distance) speed difference Δv . A possible trajectory is shown by the narrow: a vehicle, initially in free flow condition (i.e in the no reaction zone), starts approaching a vehicle in front. As long as the distance is high enough, the driver ignores the obstacles ahead, keeping his/her pace. However, when the distance becomes too small with respect to the current speed (i.e. the reaction zone is entered) the driver starts slowing down. The braking process brings the driver to match the front car speed ($\Delta v = 0$). Any increase in the speed if the ahead vehicle is then matched by the back vehicle with some delay, due to the finite reaction time of drivers ($\Delta v < 0$). For the same reason, any reduction in front car's speed is matched by the back vehicle with delay as well (the last segment of the movement curve, in the $\Delta v > 0$, unconscious area). . . .	60
4.1	CORSIM. A screenshot of CORSIM simulator..	69
4.2	VisSim example (A). Simulation of intersections with regard to design alternatives (roundabouts, unsignalized and signal-controlled, grade separated interchanges) and design, test and analysis of vehicle-actuated signal controls.	71
4.3	VisSim example (B). The same intersection under another perspective.	71
4.4	Paramics example (A).	72
4.5	Paramics example (B).	72
4.6	Vanetmobisim road topology example.	75
4.7	Vanetmobisim road topology example imported from OpenStreetMap database. The map showed in this picture ("Quarto Miglio" district in Rome) was exported in xml-format from OpenStreetMap and then imported into Vanetmobisim simulator. The red dots represent the constrained vehicles moving to across roadmap during the simulation. .	76
4.8	SUMO simulated network example. Two example networks as simulated by SUMO; left: the city of Magdeburg from the INVENT-project; right: the area around the city of Cologne.	82
4.9	Qualnet simulator. Qualnet in visual mode.	84

LIST OF FIGURES

4.10	Qualnet simulator. Qualnet in visual mode, additional example. . . .	84
4.11	Qulanet protocol stack. This picture shows the simulator protocol stack and the general functionality of each layer.	87
4.12	Qualnet protocol model. The picture shows the finite state machine representation of a protocol in QualNet. At the heart of a protocol model is an Event Dispatcher, which consists of a Wait For Event state and one or more Event Handler states. In the Wait For Event state, the protocol waits for an event to occur. When an event for the protocol occurs, the protocol transitions to the Event Handler state corresponding to that event (e.g., when Event 1 occurs, the protocol transitions to the Event 1 Handler state). In this Event Handler state, the protocol performs the actions corresponding to the event, and then returns to the Wait For Event state. Actions performed in the Event Handler state may include updating the protocol state, or scheduling other events, or both.	88
4.13	Groovenet. An example of GrooveNet simulation, where vehicles are moving in a real map with their coordinates being updated almost in realtime.	95
4.14	Groovenet. The GrooveNet simulator allows to broadcast emergency alerts or road warnings, thus analyzing the effect on the vehicles in realtime. This unique characteristic makes GrooveNet suitable both for simulations and real research testbeds.	96
4.15	TraNS simulator. Screenshot of a light version of TraNS, called TraNS Lite.	97
4.16	NCTUns simulator. The architecture of NCTUns 6.0 network simulator.	99
4.17	NCTUns simulator. The architecture of NCTUns 6.0 network protocol simulation.	100
4.18	Benefits and drawbacks. Benefits and drawbacks of several VANETs simulator.	102
4.19	Simulation time in Ns2. Simulation time in Ns2 (version 2.34) with mobile nodes ranging from 50 up to 1000 in a 2000x2000 sqm area. . . .	104
4.20	Memory memory consumption in Ns2. Memory consumption in Ns2 (version 2.34) with mobile nodes ranging from 50 up to 1000 in a 2000x2000 sqm area.	104

4.21	Simulation time in Qualnet. Simulation time in Qualnet (version 4.5) with mobile nodes ranging from 50 up to 4000 in a 2000x2000 sqm area.	105
4.22	Memory memory consumption in Qualnet. Memory consumption in Qualnet (version 4.5) with mobile nodes ranging from 50 up to 4000 in a 2000x2000 sqm area.	105
5.1	EBW - Emergency Brake Warning. The EBW alerts the driver behind by augmenting the brake-lights notification system when a preceeding vehicle performs a severe braking maneuver.	111
5.2	IVW - Intersection Violation Warning. The IVW warns the driver if violating a red light seems imminent. Other vehicles approaching the traffic lights are also warned that a close vehicle issued a warning message.	112
5.3	Network Coding. Simple example of Network Coding mechanism. . .	119
5.4	Network Coding. The figure shows the transmission of two data packets with the classical communication paradigm compared to the network coding technique. If two nodes are communicating with an intermediary node, they can benefit of this coding technique, thus optimizing the use of the channel and gaining higher throughput (in the example, nodes A and B want to communicate using node C as relay agent; while with the classical paradigm node C has to use the channel twice in order to send the flows b_1 to B and b_2 to A , with the network coding technique only transmission of flow $b_1 + b_2$	121
5.5	Network Coding benefits in data dissemination. The figure shows the number of nodes that have to be polled by the data collector to obtain all packets.	126
5.6	Network Coding benefits in data dissemination. Comparison between flooding and network coding: number of polled nodes for different numbers of transmission per node.	127

LIST OF FIGURES

5.7	Evolution of a file in a node using SPAWN protocol. (1) A car arrives in the range of a gateway, (2) starts a download, (3) download a piece of a file, (4) after getting out of range, (5) stars to gossip with its neighbors about content availability and (6) exchanges pieces of the file, thereby getting a larger portion of the file as opposed to waiting for the next gateway to resume the download.	128
5.8	CodeTorrent vs. CarTorrent. The figure shows the aggregated downloading progress (200 nodes moving with the maximum speed of 20 m/s). The number of interested nodes is 80 (which means that the <i>popularity index</i> is 40%).	131
5.9	CodeTorrent vs. CarTorrent. Impact of mobility on average download delay.	131
6.1	LT code. Functioning scheme of LT codes.	137
6.2	LT example. Decoding of a simple LT code on an erasure channel. . .	139
6.3	RAPTOR coding. Functioning scheme of RAPTOR codes.	141
6.4	RS-SD comparison. Comparison between Robust Soliton and Shokrolahi Distributions.	142
6.5	I2V vs. I2V2V The figure shows the benefit of an I2V2V approach (like in CORP) over the classical I2V, where data are disseminated only by APs.	149
6.6	CORP Algorithm. The diagram shows the receiving phase of the CORP algorithm.	150
6.7	Coding in CORP. The figure shows the encoding symbols generation process in CORP protocol.	151
6.8	AP to MN communication. The figure shows the sequence diagram of the Access Point to Mobile Node communication.	155
6.9	MN to MN communication. The figure shows the sequence diagram of the communication among mobile nodes in CORP.	156
6.10	CORP-TCP. In CORP-TCP every mobile node has to keep track of the downloaded fragments. Each mobile node saves the first and the last fragment missing (that is the missing interval). The figure shows an example of a 10 fragments resource.	157

6.11	CORP. Cumulative success decoding probability distribution as a function of the excess received packets ratio with 8000 source packets of 64 Bytes.	160
6.12	CORP delivery ratio. The figure shows the delivery ratio as a function of the number of cooperative MNs in the simulated area after 15 minutes.	160
6.13	CORP delivery ratio. The figure shows the delivery ratio as a function of the time for different numbers of collaborating nodes. Three density scenarios are considered with respect to the number of mobile nodes: $MNs = 4$ (dotted lines), $MNs = 20$ (dashed lines) or $MNs = 80$ (continuous lines).	161
6.14	Number of Received Sets of ESs. The figure shows the total sets' number owned by each mobile node).	163
6.15	Average Set size in CORP. The figure shows the average set size over the number of mobile nodes. The set size decreases as the number of nodes increases, which improves the overall dissemination speed. . . .	164
6.16	CORP vs. CORP-TCP delivery ratio. The figure shows the delivery ratio trend for CORP and CORP-TCP.	164
6.17	Aerial view of the simulation area. The figure shows the Google Earth aerial view of the simulation area. The related map in (Open Street Map) has been subsequently cropped and imported in Vanetmobisim in order to generate realistic mobility traces.	165
6.18	Imported map for mobility trace. The figure shows the map of the simulation area (it is a real map which includes two districts, " <i>Statuario</i> " and " <i>Quarto Miglio</i> ").	167
6.19	Realistic mobility pattern. The figure shows the mobility pattern generated on a real map, according to the IDM-LC mobility model (The red dots represent the constrained vehicles moving across roadmap during the simulation.). This trace has been used to compare CORP and CORP-TCP performances.	168
6.20	Realistic mobility pattern - Detailed view. The figure shows the situation of the mobility pattern generated on a real map, according to the IDM-LC mobility model (The red dots represent the constrained vehicles moving across roadmap during the simulation.).	168

LIST OF FIGURES

6.21	CORP vs. CORP-TCP delivery ratio with a realistic mobility model (IDM_LC). The figure shows the delivery ratio trend for CORP and CORP-TCP; simulations have been performed under a realistic mobility model such as the IDM_LC.	169
------	---	-----

List of Tables

4.1	Traffic level features in SUMO/Move/TraNS, VanetMobiSim and NCTuns mobility simulators.	101
4.2	Motion level features in SUMO/Move/TraNS, VanetMobiSim and NCTuns mobility simulators.	101
4.3	Other features in SUMO/Move/TraNS, VanetMobiSim and NCTuns mobility simulators.	102
5.1	Eight high priority vehicular safety applications as chosen by NHTSA (71) and VSCC (<i>Vehicle Safety Communications Consortium</i>), which is a consortium born in May, 2002 with the mission to facilitate the advancement of vehicle safety through communication technologies (the members are: BMW, DaimlerChrysler, Ford, GM, Nissan, Toyota, and VWGM, VW). Communication frequencies ranges from 1-50 Hz and the maximum range span from 50-300 meters. Note that P2M represents "point-to-multipoint", I2V "infrastructure-to-vehicle" and V2I "vehicle-to-infrastructure".	113
5.2	A subset of the SAE common message set (more than 70 data elements) and their usage in vanet safety applications.	133
6.1	Main characteristics of an LT code.	138
6.2	Comparison of different coding schemes used to approximate a digital fountain code.	143
6.3	Example of a Connection table used in CORP-TCP.	158
6.4	Example of a Service table used in CORP-TCP.	158
6.5	CORP simulation parameters used in Qualnet.	162

LIST OF TABLES

6.6 CORP and CORP-TCP simulation parameters used in Qualnet with the IDM.LC mobility pattern.	166
--	-----

Chapter 1

Introduction

In this thesis we cope with the problem of data dissemination in vehicular ad-hoc networks (VANETs), trying to give an overview of the main techniques used in this field and presenting also a novel protocol for data dissemination called "CORP", *Cooperative Rateless Protocol*.

CORP (19) is a very innovative protocol as fully implements the theory Fountain Codes technique to quickly and effectively disseminate contents across vehicles in a wide area. Furthermore, it represents the first successful application of the said coding theory to the vehicular field which totally complies with the *infrastructure-to-vehicle-to-vehicle communication paradigm (I2V2V)*.

The concept of data dissemination is wide and meaningful, and within this work we generally refer to it as the process of spreading some amount of data over a distributed wireless network, which is a superset of a VANET.

Data exchanging on the roads is becoming more and more interesting, as the number of vehicles equipped with computing technologies and wireless communication devices is poised to increase dramatically. Communications between vehicles and within the same vehicle (inter-vehicle, or InV) is becoming a promising field of research and we are moving closer to the vision of *intelligent transportation systems (ITS)*, which can enable a wide range of applications, such as collision avoidance, emergency message dissemination, dynamic route scheduling, real-time traffic condition monitoring and any kind of "infotainment" information spreading (i.e. movies, gaming and advertisement). However, it is extremely important to consider several aspects when approaching to

1. INTRODUCTION

any kind of data transfer in a VANET, because nodes are not fixed but can move. Furthermore, in this scenario, other complications can easily arise because, unlike the well known *mobile ad-hoc networks* (MANETs), where nodes can freely move in a certain area, in VANETs, vehicles' movements are constrained by streets, traffic and specific rules.

The following are only some of the several issues which VANETs are affected by:

- *High mobility:*

the environment in which vehicular networks operate is extremely dynamic, and includes extreme configurations: in highways, relative speed of up to 300 km/h may occur, while density of nodes may be 1-2 vehicles per kilometer in low busy roads. Because of the relative movement of the vehicles, the connectivity among nodes could last only few seconds, and fail in unpredictable ways.

- *Partitioned networks:*

vehicular ad hoc networks will be frequently partitioned. The dynamic nature of traffic may result in large inter-vehicle gaps in sparsely populated scenarios, and in several isolated clusters of nodes. The degree to which the network is connected is highly dependant on two factors, such as the range of wireless links and the fraction of participant vehicles, since only a fraction of vehicles on the road could be equipped with wireless interfaces. Maintaining end-to-end connectivity, packet routing, and reliable multi-hop information dissemination will become extremely challenging in such networks.

As it concerns specifically the data transmission, in VANETs there are several additional issues to be taken into account:

- the *fading effect of the signal*, which becomes more significant due to the surrounding buildings;
- the *strong interference and collisions* related to the high number of mobile transmitters (vehicles);
- the *flapping links*, caused by both fading effect and vehicles' speed.

Furthermore, while traditional vehicular networks rely on specific infrastructures, such as roadside traffic sensors reporting data to a central database, or cellular wireless

communication between vehicles and a monitoring center, we want to focus our effort on completely decentralized data dissemination solutions, in order to avoid expensive infrastructures and increase the overall scalability of the system. In fact, how to exchange traffic information among vehicles in a scalable fashion is really an important problem to be solved in VANETs.

The main goal of this dissertation was to assess the validity of the I2V2V communication scheme for data dissemination through CORP, a data dissemination protocol mainly based on Fountain Codes (FCs) and developed at application layer on the top of UDP transport protocol. By relying on UDP at lower layer, CORP showed great performances and high delivery ratio, as it was not affected by some peculiarities of TCP such as the slow start/restart mechanism, the three way handshake and the four way connection close, which both have been proved to perform poorly in vehicular networks.

Simulation results showed that CORP performs very well compared with other two simple data dissemination approach and with a similar protocol specifically built on the top of TCP (CORP-TCP).

1.1 Layout of the Thesis

In this section we give a brief insights on this research work by showing the organization of the other Chapters. In detail, we have:

- **Chapter 2: Preliminary concepts**

This chapter is meant to introduce the basic concepts behind the emerging area of vehicular networks and data exchange, such as Ad-Hoc networks, Mobility issues, Mobile Ad-Hoc Networks (MANETs) and, at the same time, to provide an overview of the new technologies and standards for car communication systems.

- **Chapter 3: Mobility models in vehicular networks**

This Chapter describes several mobility models used to simulate vehicular ad-hoc networks.

- **Chapter 4: Vehicular network simulators**

In this chapter we presents an overview of the most popular vehicular network simulators and mobility simulators, according to their integration components.

1. INTRODUCTION

- **Chapter 5: Data dissemination survey**

This chapter surveys the different types of information exchange adopted in vehicular networks with common practices and methodologies that have been considered in research literature (e.g.: opportunistic exchange of resources between vehicles, vehicle assisted data delivery, cooperating downloading of information, etc.) with a special emphasis to the network coding technique.

- **Chapter 6: Data dissemination with rateless codes**

This is the main Chapter, where we briefly survey the fountain codes (FCs) and then introduce CORP, a novel protocol for data dissemination built on the top of UDP and FCs.

The new approach is explained in detail and the main experimental results are presented (CORP is compared to V2V, I2V and I2V2V data dissemination paradigms). Simulations have been performed with *Random Waypoint* model (RWM) and the promising *Intelligent Driver Mobility model with Lane Changes* (IDM.LC), according to which mobile nodes' movements are really close to the reality (in addition, intersections are also regulated by traffic lights).

- **Chapter 7: Conclusions and future work**

This Chapter highlights the main benefits of such an approach presented in Chapter 6 and describes the future works.

Keywords: Data dissemination in vehicular networks, vehicular networks, VANETs, Fountain Codes, Data Coding in VANETs, VANET simulation.

Chapter 2

Preliminary concepts

In the recent decade, control systems for the automotive industry have moved from the analog to the digital domain. Networked Electronic Control Units (ECUs) are increasingly being deployed in automobiles to realize diverse functions such as engine management, air-bag deployment, and even in intelligent brake systems. At the same time, emerging vehicular networks in the forms of *Intra-Vehicle (InV)*, *Vehicle-to-Vehicle (V2V)*, and *Vehicle-to-Infrastructure (V2I)* communications are fast becoming a reality and will enable a variety of applications for safety, traffic efficiency, driver assistance, as well as infotainment to be incorporated into modern automobile designs. Thus, with vehicular networks fast becoming commonplace, critical data is being exchanged with-inside and with-outside vehicle via vehicular networks, and new technologies have been developed for vehicular networks. This chapter is meant to introduce the basic concepts behind the emerging area of vehicular networks and data exchange, such as Ad-Hoc networks, Mobility issues, Mobile Ad-Hoc Networks (MANETs) and, at the same time, to provide an overview of the new technologies and standards for car communication systems.

2.1 Mobile Ad-hoc NETWORK (MANET)

In circumstances where mobile telephony as we know it is not possible or difficult, perhaps internet technology can be of help. The dependancy on a costly telecommunication infrastructure could thereby be decreased, which would be quite welcome considering the current situation in the telecom world. The technology that is to make this possible

2. PRELIMINARY CONCEPTS

is MANET, or *Mobile Adhoc Networking*.

Mobile Ad Hoc Networks (MANETs) are wireless mobile nodes that cooperatively form a network without infrastructure. Because there is no coordination or configuration prior to setup of a MANET, there are several challenges which include: routing packets in an environment where the topology is changing frequently, wireless communications issues, and resource issues such as limited power and storage. The leading way to research solutions to these difficult MANET challenges is simulation, which is discussed in more detail in chapter 3.

In a MANET, the quality or even availability of a mobile connection strongly relies on the position of the receiver relative to the field of other antennas. We can think of such a network like a honeycomb, where every antenna covers one of the cells. Almost by definition, a network like that has weak spots once it is built, because in the real world it has to coexist with geographical hinderances as forests, roads and rivers. In some areas it is not even realistic to try to build such a network, if only because it is impossible to supply the required amount of energy to the antennas (not to mention the uplink to the telecom infrastructure). Even satellite telephony - though very useful in certain situations - is not the definite answer to this problem, as the capacity is limited and there are situations (for instance underground or up in the air) where the phone cannot 'see' the satellite. Here we come to the point where we can learn a number of things from the internet: when two computers are communicating with each other on the internet, neither one is more important than the other. Well, in a cellular network the phone is subservient to the network: even if two phones are used right next to each other, any connection they make is always via the network and the field antennas and, yet, a mobile phone is as much a broadcast device as it is a receiver (we can talk and listen at the same time). A second useful attribute is the aspect of *flexibility of the connections* for information flows: nodes on the internet (routers) continually learn what connections other nodes are available. With a mobile network perhaps this information exchange should happen at a higher frequency because the individual nodes are free to move, and the availability of connections would therefore vary faster, but the method is still valid.

Now we should get a reasonably good idea of how MANET will eventually be used: mobile devices that form a peer-to-peer network for the exchange of data, or channel through speech or data to the telecom network when there is no other route available.

Furthermore, this technology could generate interesting new business models, because the end users could under certain conditions bypass commercial networks (as when you are standing next to each other). For filesharing and other peer-to-peer services MANET can surely add something to the possibilities of the currently available technologies such as 3G (UMTS and by products). Obviously, MANet itself as a technology is still young and its final impact strongly relies on the way in which both the world of telecommunications and the hardware suppliers will act on its appearance on the scene.

2.2 Vehicular Ad-hoc NETWORK (VANET)

The basic concept of VANET is straightforward: take the widely adopted and inexpensive wireless local area network (WLAN) technology that connects notebook computers to each other and the Internet, and, with a few tweaks, install it on vehicles. Of course, if it were truly that straightforward, the active VANET research community would likely have never formed and this thesis would have never been written.

If vehicles can directly communicate with each other and with infrastructure, an entirely new paradigm for vehicle safety applications can be created. Even other non-safety applications can greatly enhance road and vehicle efficiency. Second, new challenges are created by high vehicle speeds and highly dynamic operating environments. Third, new requirements, required by new safety-of-life applications, include new expectations for high packet delivery rates and low packet latency. Further, customer acceptance and governmental oversight bring very high expectations of privacy and security. Even today, vehicles generate and analyze large amounts of data, although typically this data is self-contained within a single vehicle and with a VANET, the horizon of awareness for the vehicle or driver drastically increases. Communication in VANETs can be either done directly between vehicles as one-hop communication, or vehicles can retransmit messages, thereby enabling the so called multihop communication. In order to increase coverage or robustness of communication, relays at the roadside can be deployed. Roadside infrastructure can also be used as a gateway to the Internet and, thus, data and context information can be collected, stored and processed somewhere (e.g. the upcoming *Cloud Computing*¹). It warrants repeating that the interest in ve-

¹Cloud computing is a way of computing, via the Internet, that broadly shares computer resources instead of using software or storage on a local PC. A technical definition is "*a computing capability that provides an abstraction between the computing resource and its underlying technical architecture*

2. PRELIMINARY CONCEPTS

hicular inter-networks is strongly motivated by the wealth of applications that could be enabled. First of all, active safety applications, i.e., accident prevention applications, would benefit from this most direct form of communication. Second, by collecting traffic status data from a wider area, traffic flow could be improved, travel times could be reduced as well as emissions from the vehicles. As it was concisely stated as the tenet of the Intelligent Transportation System World Congress in 2008: save time, save lives. The application classes Safety and Efficiency can be used to classify applications based on their primary purpose (Cfr. chapter 4, "Data dissemination survey"). However, the aspects of safety and efficiency cannot be seen as completely disjoint sets of features. Obviously, vehicle crashes can lead to traffic jams.¹ A message reporting an accident can be seen as a safety message from the perspective of near-by vehicles. The same message can be seen by further-away vehicles as an input to calculate an alternative route within a transport efficiency application.

While being conceptually straightforward, design and deployment of VANET is a technically and economically challenging endeavour. As described in the following chapters, key technical challenges include the following issues:

- *Inherent characteristics of the radio channel.*

VANET present scenarios with unfavorable characteristics for developing wireless communications, i.e., multiple reflecting objects able to degrade the strength and quality of the received signal. Additionally, owing to the mobility of the surrounding objects and/or the sender and receiver themselves, fading effects have to be taken into account.

- *Lack of an online centralized management and coordination entity.*

The fair and efficient use of the available bandwidth of the wireless channel is a hard task in a totally decentralized and self-organizing network. The lack of an entity able to synchronize and manage the transmission events of the different nodes might result in a less efficient usage of the channel and in a large number of packet collisions.

(e.g., servers, storage, networks), enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction." - (Cloud Computing Definition, National Institute of Standards and Technology, Version 15, <http://csrc.nist.gov/groups/SNS/cloud-computing/index.html>)

2.2 Vehicular Ad-hoc NETWORK (VANET)

- *High mobility, scalability requirements, and the wide variety of environmental conditions.*

The challenges of a decentralized self-organizing network are particularly stressed by the high speeds that nodes in VANET can experience. Their high mobility presents a challenge to most iterative optimization algorithms aimed at making better use of the channel bandwidth or the use of predefined routes to forward information.

- *Security and privacy.*

There is a challenge in balancing security and privacy needs. On the one hand, the receivers want to make sure that they can trust the source of information. On the other hand, the availability of such trust might contradict the privacy requirements of a sender.

- *Standardization versus flexibility.*

Without any doubt, there is a need for standardizing communications to allow VANET to work across the various makes and brands of original equipment manufacturers (OEMs). Yet, it is likely that OEMs will want to create some product differentiation with their VANET assets. These goals are somewhat in tension.

From an application and socio-economic perspective, key challenges are as follows:

- Analyzing and quantifying the benefit of VANET for traffic safety and transport efficiency. So far, relatively little work has been done to assess the impact of VANET as a new source of information on driving behavior. Clearly, the associated challenge in addressing the issue of impact assessment is the modelling of the related human factor aspects.
- Analyzing and quantifying the cost/benefit relationship of VANET. Because of the lack of studies on the benefits of VANET, a cost/benefit analysis can hardly be done.
- Designing deployment strategies for this type of VANET that are not based on a single infrastructure and/or service provider. Owing to the network effect, there is the challenge of convincing early adopters to buy VANET equipment when they will rarely find a communication partner.

2. PRELIMINARY CONCEPTS

- Embedding VANET in intelligent transportation systems architectures. VANET will be a part of an intelligent transportation system where other elements are given by traffic-light control or variable message signs. Also public and individual transportation have to be taken into account in a joint fashion. Therefore, truly cooperative systems need to be developed.

As can be seen from the above lists of technical, application, and socio-economic aspects, the field of vehicular application and inter-networking technologies is based on an interdisciplinary effort in the cross section of communication and networking, automotive electronics, road operation and management, and information and service provisioning. VANET can therefore be seen as a vital part of Intelligent Transportation Systems (ITS).

2.3 Network requirements in VANETs applications

This section aims to be a very light introduction to the main technological requirements of vehicular networks which, in most of the cases, need to cover many needs for their efficient operation. In this section we will treat only the most important ones.

2.3.1 Mobility

Wireless network technologies allow devices to move freely. However, this mobility affects the potential permanent access to the network (see the previous point) and causes other problems. In (102), experimental evaluations give real results of these effects. In 802.11 transmissions the distance between the sender and receiver is an important factor; the more the distance, the smaller the probability of reception of packets (24). In infrastructure-based technologies, handoffs between base stations are also relevant, due to the potential decrease of performance in the process. Poor latency and throughput results are obtained if the mobile terminal is moving at locations far away from other nodes without performing a handoff (42). Nevertheless, the distance between two devices during the communication is not the only noticeable effect of mobility. Interference with other radio equipments in the case of VANET should also be taken into account, due to the wide usage of the 2.4 GHz frequency band (102). The presence of the equipment at locations of bad orography could also cause communication problems in vehicular networks. Other external factors, like the existence of other

2.3 Network requirements in VANETs applications

vehicles or buildings are considered in realistic mobility patterns for VANET solutions (69).

Mobility will be examined in more detail in chapter 2 ("Mobility models in vehicular

Applications\App. Req.	Location awareness	Geocast capability	Penetration rate dependence	Time awareness	Permanent access	Mobility
Safety						
Cooperative Collision Warning	**	**	**	**	*	**
Incident management	**	**	**	**	*	**
Emergency video streaming	**	**	**	**	*	**
Traffic management and monitoring						
Platooning	**	**	**	**	*	**
Vehicles tracking	*		*	*	**	*
Notification Services	*		*	*	*	*
Comfort						
Parking place management	**	**	*	*	*	**
Distributed games and/or talks	*	**	**	*	*	**
Peer-to-peer	*		*	*	**	*

Figure 2.1: Application requirements in VANETs. Please, note that no asterisk means "none or not needed" requirement, "*" means "needed", "***" stands for "suited" requirement.

networks") and chapter 3 ("Vehicular Network Simulators").

2.3.2 Permanent access

Permanent access to the network is *one of the main drawbacks of vehicular communications*. In VANET designs, a physical infrastructure is not necessary, due to the inherent decentralized design. As it concerns infrastructure-based networks, operators do not offer the same service over the entire terrestrial surface. For instance, over urban environments, the coverage is excellent, and the amount of base stations where the mobile terminal could be connected is really high. At rural locations, however, the deployment is poor. A vehicle equipped with a VANET system, however, is always able to emit messages because the vehicle itself is part of the infrastructure. Furthermore, in cellular network connections, it is also important to differentiate between two important concepts regarding the access to the network: *coverage* and *capacity*. The coverage can be understood as the possibility of the mobile terminal to use the network, because at a particular location operators have deployed the necessary infrastructure. However, the user can be rejected to establish a call or a data connection, even in good coverage circumstances, if the capacity of the network has been exceeded. Depending on several

2. PRELIMINARY CONCEPTS

technological issues, such as modulation, frequency allocation, time slot scheduling, etc., this effect has a different impact. According to this, the number of users who are concurrently using the network restricts the potential cellular network usage. At the application level, some services such as file transfer or download, need a permanent communication channel. In this kind of applications, the election of a suited vehicular network is essential.

2.3.3 Location Awareness

Next generation vehicles are expected to exchange information not only beyond their immediate surroundings and line-of-sight with other vehicles, but also with the road infrastructure and Internet databases. This will allow vehicles to anticipate trajectories, coordinate merging manoeuvres, notify a braking action to vehicles behind, warn oncoming traffic of an icy patch, report road traffic conditions, locate parking lots, or simply entertain passengers. In this context, the knowledge of their actual position and trajectory is necessary, and it is only meaningful to vehicles in a particular geographic area. The exchange of information among vehicles in a particular geographic area requires reliable and scalable communication capabilities, which we call geographical routing and addressing. This function mainly rely on the information given by GPS receivers. However, GPS imposes some constraints such as lack of coverage in some environments or its weak robustness for some critical applications. For these reasons, other positioning techniques such as cellular or WiFi localization, dead reckoning (by using last known position and velocity) (94), and image/video localization, have been proposed in the vehicular field (16). Critical safety services such as alert cooperative collision warning and incident management need a high accurate localization, as well as some comfort applications such as parking booking. Note that an accurate positioning system can help us to define the zone of relevance more precisely. Other services, however, require a low accurate localization, like peer to peer applications and vehicle tracking.

2.3.4 Time Awareness

Vehicular applications often require a reliable communication channel that supports time-critical message transmissions (41). One of the most important criterions for

measuring the quality of the network, regardless of the application type, is the communication delay. Although most applications have time constraints, those related with road safety are critical. Due to this, a challenge in vehicular networks is providing a real-time behaviour. In order to enable the driver to react quickly, the information must reach the destination in a very small delay following the event. However, this requirement is not easy to ensure in mobile networks. This difficulty is even greater if we consider vehicular network characteristics, particularly, the high mobility. Thus, real-time communications can only be assured by the presence of an efficient and robust communication system.

2.3.5 Penetration rate dependency

Penetration rate is defined as the percentage of vehicles equipped with the necessary *on board data unit (OBU)* on the road. This parameter may have important consequences in the operation of some applications, especially the critical and safety ones. Although a low penetration rate is obviously a problem in safety applications, such as collision avoidance, an excess of equipped vehicles also arises transmission problems. However, applications such as comfort do not have to be too much aware of this factor. In cellular networks, situations of high penetration are also a problem. The system performance is not affected when the number of equipped vehicles is low, but in high load circumstances, the network connection starts to give a poor performance when the time slot scheduler need to serve too much users (52). Note that penetration rate has a direct bearing on the wireless bandwidth used. The higher the penetration rate, the higher the wireless bandwidth should be used to allow vehicles to communicate.

2.3.6 Geocast capability

Geocast provides the capability to deliver a message to nodes within a geographical region (62). The shape and size of this area depend on the application aims. The complexity of defining this region can be as high as the set of vehicles behind or in front of the subject one. Other times this constraint is relaxed, and defining this region as the vehicles inside a geographic area, or near a designated spot (such as a smog area), is enough. In order to advocate a general communication architecture, where services which require both unicast and geocast capabilities can be deployed, an hybrid networking architecture can be proposed. This way, services such as platooning, which needs

2. PRELIMINARY CONCEPTS

Technology	Range	Link type	Data rate	Frequency band	Standard	Vehicular applicability		
						V2V	V2I	I2V
Bluetooth	100 m	1-to-n	1 Mbps	2.4 Ghz	IEEE 802.15.1	*		
WLAN	200 m	1-to-1 1-to-n	10-50 Mbps	2.4,5 Ghz	IEEE 802.11a/b/g	**	*	*
DSRC	1 Km	1-to-1	50 Mbps	5.9 Ghz	IEEE 802.11p	**	**	**
WiMAX	10 Km	1-to-n	~20 Mbps	2.4,5 Ghz	IEEE 802.16e		**	**
Cellular	10 Km	1-to-n	~10 Mbps	700-2600 Mhz	n/a		**	**
RDS/TMC	80 Km	1-to-n	1187.5 bps	87.5-108.0 Mhz	CENELEC EN 50067 CEN ENV 12313			**
Satellite	>10.000 Km	1-to-n	300-500 Kbps	950-1450 Mhz	n/a		*	**

Figure 2.2: Geocasting forwarding schemes. Typical geocasting forwarding schemes: *unicast* (A), *broadcast* (B) and *topologically scoped broadcast* (C).

unicast communications, do not experience bad performances. Geocast is considered efficient if the information is forwarded in both sparse and dense geographical areas, while efficiently leveraging the available bandwidth. This criterion, scalability, was introduced in (70), and it was defined as the ability to handle the addition of nodes or objects without suffering a noticeable loss in performance or increase in administrative complexity.

2.4 VANETs' common communication schemes

Connectivity necessities of vehicles can be divided in two main groups: vehicle to *vehicle communications* (V2V) or *inter-vehicle communications* (InV), and *communications with the infrastructure* (V2I). In the literature, many authors use V2I to denote both data flow directions, however, according to the specific use of several technologies for one or the other communication pattern, it is more correct to distinguish between V2I and I2V (infrastructure to vehicle communications). It is important to consider this whole set of communication possibilities for vehicles because, depending on the application or service necessities, we will have to decide among one of the available wireless network technologies. Apart from the communication pattern covered, wireless communication technologies can be divided into those which establish 1-to-1 physical links, and those which consider 1-to-n broadcast ones. In this last case, some kind of access point is in charge of sharing out the available bandwidth among the clients. This bandwidth, thus, could become insufficient when the number of served nodes increase inside the coverage area. Due to this, the tendency in short-range wireless technologies is taking advantage of the available bandwidth, sharing it among a small number of

2.4 VANETs' common communication schemes

users because, anyway, the coverage is small. On the contrary, wide-range technologies must share the available bandwidth among much more users. However, short-range wireless media lack on stability, due to the small accessible area. It is also important to remark how V2V communications are obtained by means of 1-to-1 technologies, and communications with the infrastructure are commonly created using the 1-to-n ones. A brief overview of main wireless technologies used in the vehicular domain is given in Table 2.6. For communications with the infrastructure, it is said that WLAN, DSRC, WiMAX, cellular and satellite are feasible. However it is important to remark the different application they cover in this domain. In the case of WLAN/DSRC, vehicles usually connect with local roadside units, what usually is called vehicle to road side communication.

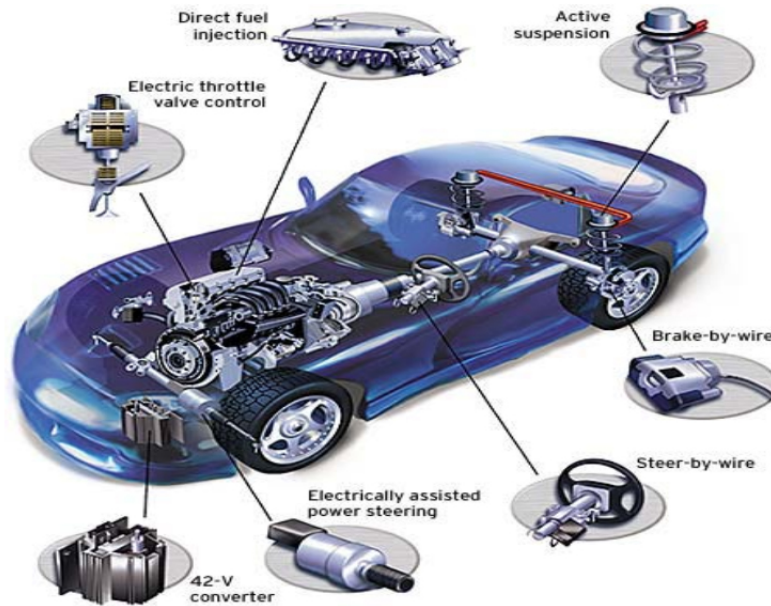


Figure 2.3: InV communications. The InV communication scheme is mainly used to create a specialized sensor networks within the vehicle. It is very useful when you want to control vehicle's on board devices and, at the same time, act on the motion behaviour in response to a received emergency message. (Mercedes Benz, SL 500 - source: Daimler Benz)

On the other hand, in the WiMAX/cellular case it is used a medium range 1-to-n network, and in satellite communications a wide range 1-to-n model is applied. In the

2. PRELIMINARY CONCEPTS

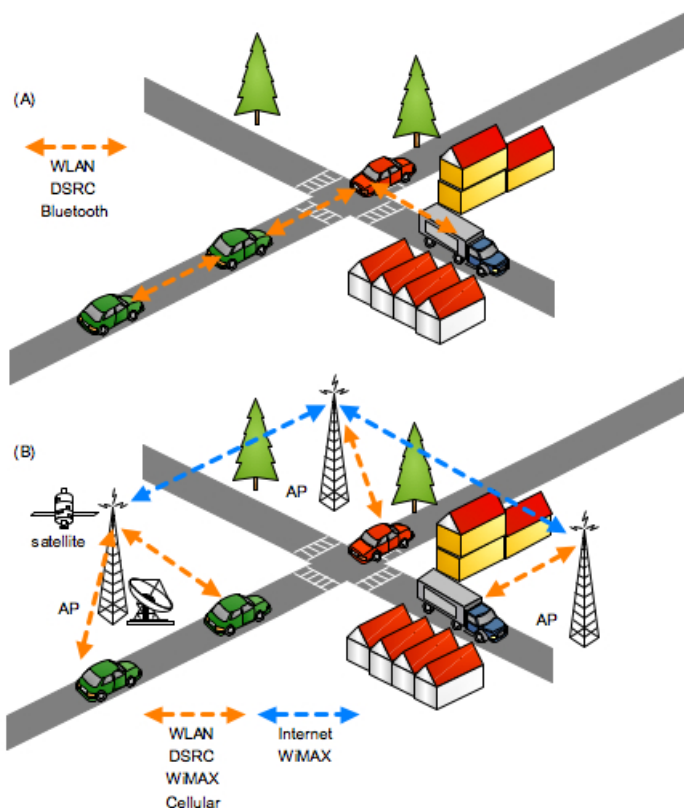


Figure 2.4: Communication schemes in VANETs. The figure shows the communication schemes typically used in a VANET: (a) Vehicle-to-Vehicle (V2V), (b) Vehicle-to-Infrastructure (V2I) and Infrastructure-to-Vehicle (I2V).

cellular and satellite cases the design of the network is even more fixed than with any other technology, because we have to use the operators installations. This way, service providers usually consider the direct Internet connection offered by the operator, and there is no possibility to manage data traffic inside the operators network.

2.5 Communication technology in VANETs

With the rapid development of information technologies, there are a number of wireless technologies which are potential for wireless InV, V2V and V2I communications, (cfr. Table 2.2). These new technologies could be used for data exchange between users devices and vehicles, among vehicles, and between vehicles and infrastructure. The details of application scenarios of data exchange with each technology are listed in two

tables as well. DSRC is the recent technological trends to provide real time traffic information for effective implementation of ITS. Thus, in the following subsection, we focus on introducing the new technologies of DSRC for vehicular networks.

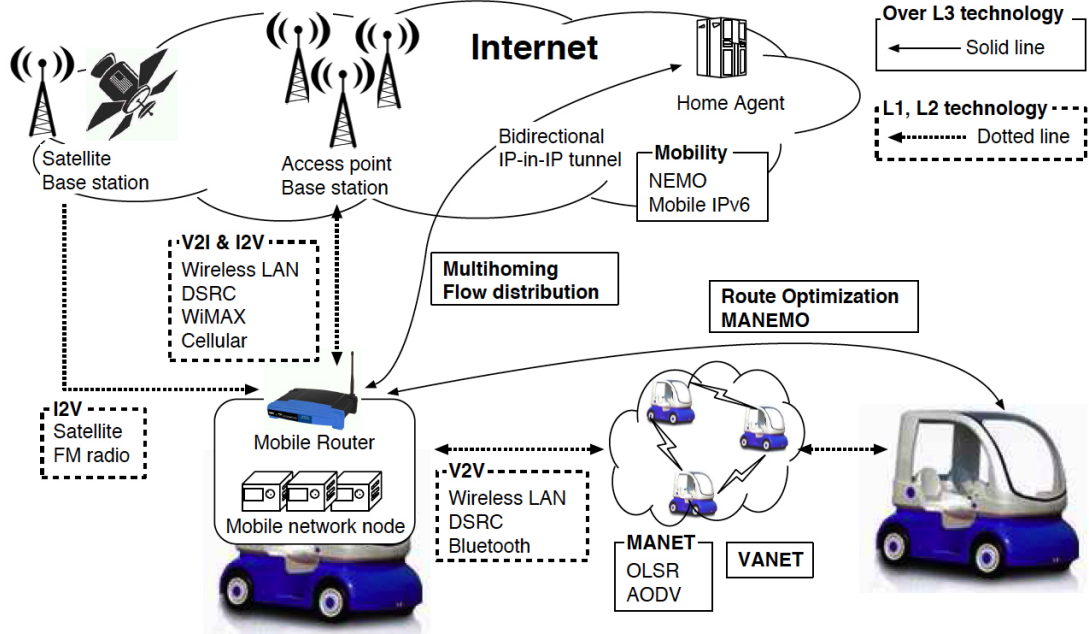


Figure 2.5: Overview of the application of network technologies used in vehicle ad-hoc networks. The figure shows a vehicular networking scheme where the most important technologies at level-three are included in an integral communication solution. Finally, an overlay architecture using cellular networks shows the feasibility of this technology to enable vehicular communications.

2.5.1 WLAN - IEEE 802.11

At the end of the 1990s the first devices appeared on the market using a new wireless local area network technology that is commonly referred to today as WLAN (Wireless LAN) or Wi-Fi.

Wi-Fi is specified by the IEEE in the 802.11 standard and maintained by the IEEE LAN/MAN Standards Committee (IEEE 802)¹: it is very similar to the 802.3 fixed line Ethernet standard and reuses all protocol layers down to layer 2. The major difference between the two protocols is on layer 1, where the fixed line medium access has been

¹<http://www.ieee.org>

2. PRELIMINARY CONCEPTS

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

Figure 2.6: Wireless technologies for InV communications.

replaced with several wireless variants. Furthermore, some additional management features were specified that address the specific needs of wireless transmissions that do not exist in fixed line networks, such as network announcements, automatic packet retransmission, authentication procedures and encryption. Over the years, several physical layer standards were added to increase transmission speeds and to introduce additional features. Devices are usually backwards-compatible and support all previous standards to enable newer and older devices to communicate with each other.

Initially, Wi-Fi was not very popular or widely known as network interface cards were expensive and transmission speeds ranged between 1 and 2 Mbit/s. Things changed significantly with the introduction of 802.11b, which specified a physical layer for transmission speeds of up to 11 Mbit/s. Network interface cards became cheaper and devices appeared that could be connected to PCs and notebooks over the new high-speed USB (Universal Serial Bus) interface. Prices fell significantly and Intel decided to include Wi-Fi capabilities in their Centrino notebook chipsets. At the same time, the growing popularity of high-speed DSL and TV cable Internet connectivity made wireless networking more interesting to consumers, since the telephone or TV outlet was and still is often not close to where a PC or notebook is located. Wi-Fi was the ideal solution to this problem and Wi-Fi access points were soon integrated into DSL and cable

modems. Likewise, Internet access in public places such as cafes, hotels, airports and so on became popular, again enabled by Wi-Fi and cheap high-speed Internet access at the other end of the wireless connection via DSL. Today, Wi-Fi has become ubiquitous in notebooks and many other mobile and portable devices such as game consoles, mobile phones, Internet tablets and Mobile Internet Devices (MID).

Over time, two additional physical layer specifications were added to further increase transmission speeds. The 802.11g standard increased data transfer speeds to up to 54 Mbit/s on the air interface, and the recent 802.11n standard has the potential for up to 300 Mbit/s. It should be noted at this point that these speeds are only theoretical and not measured on the air interface. In practice, protocol overhead reduces the achievable speeds at the application layer to about half those values and, especially when applied to vehicular networks, this may lead to very poor communication efficiency. Standard 802.11a is another Wi-Fi air interface variant, but has never gained much popularity because it does not use the same standard frequency band as the other 802.11 variants. Very important topic (which are out of the scope of this work) are *Wi-Fi security* and, due to the popularity of Wi-Fi and the growing use of the technology for real-time applications such as VoIP and video streaming, *quality of service (QoS)*.

As vehicular networks make intensive use of this technology, it is worth to see in more detail the 802.11 protocol family:

802.11-legacy

The original version of the standard IEEE 802.11 was released in 1997 and clarified in 1999, but is today obsolete. It specified two net bit rates of 1 or 2 megabits per second (Mbit/s), plus forward error correction code. It specified three alternative physical layer technologies: diffuse infrared operating at 1 Mbit/s; frequency-hopping spread spectrum operating at 1 Mbit/s or 2 Mbit/s; and direct-sequence spread spectrum operating at 1 Mbit/s or 2 Mbit/s. The latter two radio technologies used microwave transmission over the Industrial Scientific Medical frequency band at 2.4 GHz. Some earlier WLAN technologies used lower frequencies, such as the U.S. 900 MHz ISM band. Legacy 802.11 with direct-sequence spread spectrum was rapidly supplanted and popularized by 802.11b.

802.11a

This standard uses the same data link layer protocol and frame format as the

2. PRELIMINARY CONCEPTS

original standard, but an OFDM¹ based air interface (physical layer). It operates in the 5 GHz band with a maximum net data rate of 54 Mbit/s, plus error correction code, which yields realistic net achievable throughput of 20 Mbit/s. Since the 2.4 GHz band is heavily used to the point of being crowded, using the relatively unused 5 GHz band gives 802.11a a significant advantage. However,

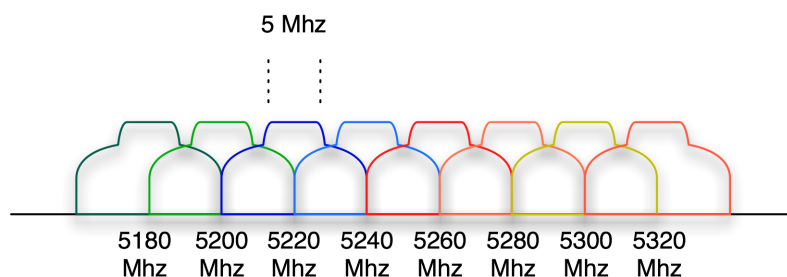


Figure 2.7: 802.11a channels.

this high carrier frequency also brings a disadvantage: the effective overall range of 802.11a is less than that of 802.11b/g. In theory, 802.11a signals are absorbed more readily by walls and other solid objects in their path due to their smaller wavelength and, as a result, cannot penetrate as far as those of 802.11b. In practice, 802.11b typically has a higher range at low speeds (802.11b will reduce speed to 5 Mbit/s or even 1 Mbit/s at low signal strengths). However, at higher speeds, 802.11a often has the same or greater range due to less interference. *In Italy it is forbidden to use 802.11a technology due to high RF interferences with militar radars and strategical devices.*

¹Orthogonal frequency-division multiplexing (OFDM) is a frequency-division multiplexing (FDM) scheme utilized as a digital multi-carrier modulation method. A large number of closely-spaced orthogonal sub-carriers are used to carry data. The data is divided into several parallel data streams or channels, one for each sub-carrier. Each sub-carrier is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate, maintaining total data rates similar to conventional single-carrier modulation schemes in the same bandwidth. OFDM has developed into a popular scheme for wideband digital communication, whether wireless or over copper wires, used in applications such as digital television and audio broadcasting, wireless networking and broadband internet access. The primary advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long copper wire, narrowband interference and frequency-selective fading due to multipath) without complex equalization filters. (source: <http://www.wikipedia.com>)

802.11b

802.11b has a maximum raw data rate of 11 Mbit/s and uses the same media access method defined in the original standard. 802.11b products appeared on the market in early 2000, since 802.11b is a direct extension of the modulation technique defined in the original standard. The dramatic increase in throughput of 802.11b (compared to the original standard) along with simultaneous substantial price reductions led to the rapid acceptance of 802.11b as the definitive wireless LAN technology. 802.11b devices suffer interference from other products operating in the 2.4 GHz band. Devices operating in the 2.4 GHz range include: microwave ovens, Bluetooth devices, baby monitors and cordless telephones.

802.11g

In June 2003, a third modulation standard was ratified: 802.11g. This works in the 2.4 GHz band (like 802.11b), but uses the same OFDM based transmission scheme as 802.11a. It operates at a maximum physical layer bit rate of 54 Mbit/s exclusive of forward error correction codes, or about 22 Mbit/s average throughput. 802.11g hardware is fully backwards compatible with 802.11b hardware and therefore is encumbered with legacy issues that reduce throughput when compared to 802.11a by $\sim 21\%$. The then-proposed 802.11g standard was rapidly

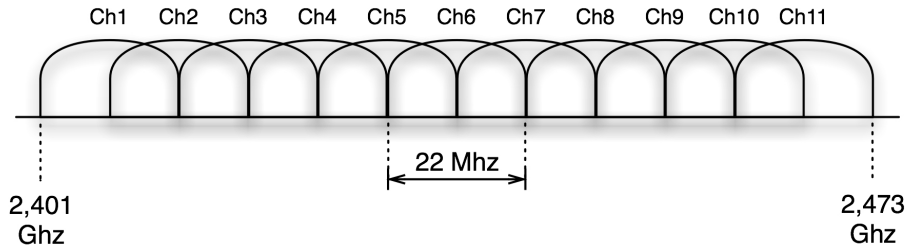


Figure 2.8: 802.11b/g channels.

adopted by consumers starting in January 2003, well before ratification, due to the desire for higher data rates as well as to reductions in manufacturing costs. By summer 2003, most dual-band 802.11a/b products became dual-band/tri-mode, supporting a and b/g in a single mobile adapter card or access point. Details of making b and g work well together occupied much of the lingering technical

2. PRELIMINARY CONCEPTS

process; in an 802.11g network, however, activity of an 802.11b participant will reduce the data rate of the overall 802.11g network. Like 802.11b, 802.11g devices suffer interference from other products operating in the 2.4 GHz band.

802.11-2007

In 2003, task group TGma was authorized to "roll up" many of the amendments to the 1999 version of the 802.11 standard. REVma or 802.11ma, as it was called, created a single document that merged 8 amendments (802.11a,b,d,e,g,h,i,j) with the base standard. Upon approval on March 8, 2007, 802.11REVma was renamed to the current base standard IEEE 802.11-2007.

802.11n - MIMO

802.11n is a recent amendment which improves upon the previous 802.11 standards by adding multiple-input multiple-output (MIMO) and many other newer features. The IEEE has approved the amendment and it was published in October 2009.⁽⁸⁾ ¹ MIMO technology offers tremendous performance gains for WLANs at relatively low cost. Any system with multiple inputs into the receiver and multiple outputs to the transmitter is a MIMO system, but implementing such a system involves several distinctly different radio techniques. Some of these techniques are beneficial and fully compatible with today's standard WLAN equipment, while others do not improve performance when used with existing equipment.

In MIMO, multiple antennas are used to coherently resolve more information than possible using a single antenna. One way it provides this is through *Spatial Division Multiplexing (SDM)*. SDM spatially multiplexes multiple independent data streams, transferred simultaneously within one spectral channel of bandwidth. MIMO SDM can significantly increase data throughput as the number of resolved spatial data streams is increased. Each spatial stream requires a discrete antenna at both the transmitter and the receiver. In addition, MIMO technology requires a separate radio frequency chain and analog-to-digital converter for each MIMO antenna which translates to higher implementation costs compared to non-MIMO systems.²

¹http://standards.ieee.org/announcements/ieee802.11n_2009amendment_ratified.html

²<https://mentor.ieee.org/802.11/dcn/09/11-09-0576-03-000n-sp2-40mhz-coexistence-cids-presentation.ppt>

40 MHz channels is another feature incorporated into 802.11n which doubles the channel width from 20 MHz in previous 802.11 PHYs to transmit data. This allows for a doubling of the PHY data rate over a single 20 MHz channel. It can be enabled in the 5 GHz mode, or within the 2.4 GHz if there is knowledge that it will not interfere with any other 802.11 or non-802.11 (such as Bluetooth) system using those same frequencies.

In order to better understand how the antennas are used in a MIMO communication system, we can say that the number of simultaneous data streams is limited by the minimum number of antennas in use on both sides of the link. However, the individual radios often further limit the number of spatial streams that may carry unique data. The notation $a \times b : c$ helps identify what a given radio is capable of. The first number (a) is the maximum number of transmit antennas or RF chains that can be used by the radio. The second number (b) is the maximum number of receive antennas or RF chains that can be used by the radio. The third number (c) is the maximum number of data spatial streams the radio can use. For example, a radio that can transmit on two antennas and receive on three, but can only send or receive two data streams would be $2 \times 3 : 2$. The 802.11n draft allows up to $4 \times 4 : 4$. Common configurations of 11n devices are $2 \times 2 : 2$, $2 \times 3 : 2$, and $3 \times 3 : 3$. All three configurations have the same maximum throughputs and features, and differ only in the amount of diversity the antenna systems provide. In addition, a fourth configuration, is becoming common, which has a higher throughput, due to the additional data stream. The 802.11n MIMO was so promising since its debut that even before the final ratification, enterprises were already migrating to 802.11n networks based on the Wi-Fi Alliance's certification of products conforming to a 2007 draft of the said technology.

The most popular standards are those defined by the 802.11b and 802.11g protocols, which are amendments to the original standard. 802.11-1997 was the first wireless networking standard, but 802.11b was the first widely accepted one, followed by 802.11g and 802.11n. 802.11n is a new multi-streaming modulation technique. Other standards in the family (cf, h, j) are service amendments and extensions or corrections to the previous specifications. Given that both 802.11b and 802.11g use the 2.4 GHz ISM band, the relevant equipment may occasionally suffer interference from microwave ovens, cordless

2. PRELIMINARY CONCEPTS

telephones and Bluetooth devices. Both 802.11 and Bluetooth control their interference and susceptibility to interference by using spread spectrum modulation. Bluetooth uses a frequency hopping spread spectrum signaling method (FHSS), while 802.11b and 802.11g use the direct sequence spread spectrum signaling (DSSS) and orthogonal frequency division multiplexing (OFDM) methods, respectively. 802.11a uses the 5 GHz U-NII band, which, for much of the world, offers at least 19 non-overlapping channels rather than the 3 offered in the 2.4 GHz ISM frequency band. Better or worse performance with higher or lower frequencies (channels) may be realized, depending on the environment. The used segment of the radio frequency spectrum varies between countries. In the US, 802.11a and 802.11g devices may be operated without a license, as allowed in Part 15 of the FCC Rules and Regulations. Frequencies used by channels one through six (802.11b) fall within the 2.4 GHz amateur radio band. Licensed amateur radio operators may operate 802.11b/g devices under Part 97 of the FCC Rules and Regulations, allowing increased power output but not commercial content or encryption. In Italy, according to the ETS 300-328-2, it is forbidden to exceed the 100 mW EIRP threshold¹, which is equivalent to 20 dBm gain. Furthermore, radio devices for private use can transmit with a maximum electrical power of 50 mW (equivalent to 17 dBm) because, usually, even the simplest antenna has a transmission gain of 2.5 dBi, which makes possible to reach an EIRP of 80 mW (equivalent to 19.2 dBm). For this reason, in European Union it is strictly forbidden to use antennas with an high gain (higher than 5 dBi), because it could be possible to raise the EIRP beyond the 100 mW threshold (20 dBm).

2.5.2 WiMAX

WiMAX, or *Worldwide Interoperability for Microwave Access*, is a communication technology which try to fill the gap between 3G and WLAN standards, and it is the first implementation which appears to comply with the MAN (Metropolitan Area Network) concept, in a wireless manner. Two main standards are currently considered: 802.16d

¹*Equivalent isotropically radiated power (EIRP)* or, alternatively, *Effective isotropically radiated power* is the amount of power that a theoretical isotropic antenna (which evenly distributes power in all directions) would emit to produce the peak power density observed in the direction of maximum antenna gain. EIRP can take into account the losses in transmission line and connectors and includes the gain of the antenna. The EIRP is often stated in terms of decibels over a reference power emitted by an isotropic radiator with an equivalent signal strength.

and 802.16e. The first one is used at fixed locations, and it is a perfect solution for connecting different buildings of a company at a low cost, for example. This specification offers up to 48 Km of coverage and data rates of 70 Mbps.

The 802.16e standard, specifically designed for mobile users connected to a base station. The OFDM (*Orthogonal Frequency Division Multiple access*) technology is used in this standard to serve multiple users, and the final physical interface considered copes with mobility issues, such as interferences, multipath and delays. 802.16e is, hence, the most appropriate specification of WiMAX for the vehicular field. Tens of Mbps, mobility speed up to 100 Km/h, and 10 km of coverage to the base station, make 802.16e a good option for urban scenarios, where vehicles can be connected at a high data rate using a WiMAX deployment.

Currently it is possible to obtain some Pre-WiMAX devices, but it is expected that, as soon as the final specifications are ready, the spectrum of vehicular services which could be deployed with WiMAX grow rapidly. This technology was stressed in order to analyze its performance in a mobile environment and the results showed that a mean of 2 Mbps and 5.3 Mbps can be obtained in real scenarios (up to 90 Km/h), with an average RTT¹ of 100 ms.

2.5.3 Bluetooth

Bluetooth technology aims at allowing wireless short-range communications between several devices. Developed originally by Ericsson, Bluetooth undergoes an evolution of its specifications maintained and developed by the Special Interest Group (SIG) of Bluetooth and is accessorially standardized by the IEEE under the reference IEEE 802.15.1. Today IEEE 802.15 subgroups and other forums such as the Wimedia Alliance are competing for the same field of operation. The governing idea behind Bluetooth consisted in specifying a wide scale integrated circuit to be deployed on a very large scale on various types of equipments with a very reduced energy consumption and thus announcing very low prices.

In 1994, Ericsson Mobile Communications launched a feasibility study of low-cost low-consumption radio interface to be used between mobile telephones and their accessories.

¹RTT stands for *Round Trip Time* and it is the time required to send a signal in both directions over a particular communication link. This is the soonest that it is possible to receive an acknowledgement of a message.

2. PRELIMINARY CONCEPTS

In February 1998, IBM, INTEL, Nokia and Toshiba joined the Swedish company and in May they created the SIG. It was widened by the arrival of 3Com, Agere (Lucent Technologies), Microsoft and Motorola during 2000 and does not cease increasing thus gathering actors who cover several elds of expertise such as cellular telephony, portable computers, cars and digital processing. Being an opened industrial specification, all the members of Bluetooth SIG can use it free in their products and services. Today, the SIG counts more than 2500 manufacturers. It carries out a true ght to promote this world standard despite the large number of concurrent technologies essentially headed by Wi-Fi.

Thanks to Bluetooth, however, it is possible to create a personal area network (PAN) where several devices can be connected. It operates in the 2.4 GHz band and, due to the low power consumption features, allow communications in a typical range of tens of meters. Bluetooth terminals are grouped in piconets, and these piconets can also be connected by means of scatternets.

The properties of Bluetooth make it perfect for invehicle networks (72). Some researchers also advocate the usage of Bluetooth for V2V applications (92). However, this technology is limited by the necessary time to form piconets and scatternets (in the order of seconds) (86) and, overall, the limited communication range.

2.5.3.1 Bluetooth Architecture

Bluetooth communication requires two preliminary things: rst we have to know the devices in the neighbourhood (discovery) and second there must be a preestablished circuit. Communication is also based on a masterslave principle. A group of equipments forms a cell called piconet. A piconet comprises a master and seven slaves at the maximum. Several piconets can overlap and form a *scatternet* (see Fig.2.9). In a piconet the communication is based on the master to harmonize the frequencies and channels. We know the neighbours through the discovery phase while in a scatternet there is a need to route data between masters and relay nodes. Scatternets in Bluetooth is not well developed. It has been improved by specic routing procedures in later standards such as ZigBee. Two slave devices cannot talk directly to each other except during the discovery phase. Channel allocation and communication establishment are under the responsibility of the master. Although there was a limitation in earlier versions of Bluetooth on the number of simultaneous channels in a piconet, it is removed from the

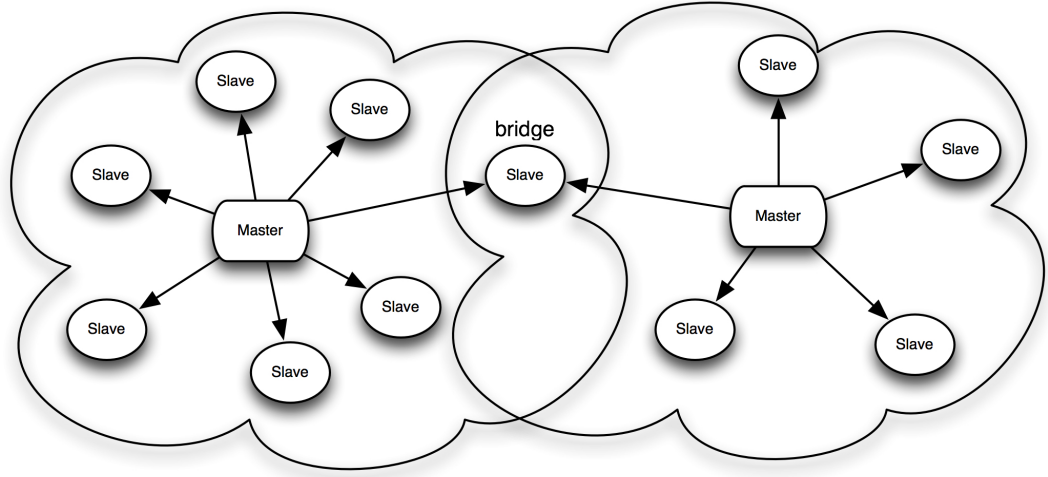


Figure 2.9: Example of a bluetooth scatternet.

current version as the cell capacity has increased significantly. The standard supports also broadcast by simply removing the destination from the messages. The master is responsible of polling nodes and also allocating/blocking new connection bandwidth. It is responsible for setting the piconet synchronization clock and as we will see decides for the *frequency hopping sequence (FHS)*. A slave can be part of several piconets. One major interesting feature of Bluetooth is that it is not dependent on the IP. This courageous design decision eases the deployment of devices that do not need to worry about upper layer problems such as address allocation, default router, netmask, etc. Auto configuration is hence much easier.

2.5.4 ZigBee

This is an In-Vehicle (InV) wireless sensor network which offers lower data rate but can be used for many embedded applications. ZigBee is an industry alliance that promotes a set of rules which builds on top of the IEEE 802.15.4 standards; it is a low-cost, low-power, wireless mesh networking standard. It operates in three ISM radio bands: 868 MHz in Europe, 915 MHz in countries such as USA and Australia, and 2.4 GHz (16 channels) in most jurisdictions worldwide. The modulation uses direct-sequence spread spectrum coding, which is managed by the digital stream into the modulator

2. PRELIMINARY CONCEPTS

and the data rate is 250 Kbps per channel in the 2.4 GHz band (40 Kbps in the 915 MHz band and 20 Kbps in the 868 MHz band). Transmission range is between 10 and 75 meters, although it is heavily dependent on the particular environment. The maximum output power of the radios is generally 0 dBm (1 mW). CSMA/CA is the usually channel access mode, except three cases (beacons, acknowledgment, messages for beacon oriented network devices).

As ZigBee provides network speeds of up to 250 Kbps, it is expected to be largely used as a sensor network for monitoring and control purposes (air conditioning, heating, ventilation, lighting control, etc.).

In general, the experiments and measured results indicate that ZigBee is a viable and promising technology for implementing an intra-car wireless sensor network (10). However, the communication between sensor nodes and a base station in the car will depend on several factors, such as the power loss, coherence bandwidth, and coherence time of the underlying communication channels between the sensor nodes and the base station. Other high data rate technologies are available such as IEEE 802.15.3a (UWB) which is a competitor to the other IEEE 802.11 solutions. It can provide speeds of up to hundred of Mbps/s and it can be likely used in multimedia applications. However, UWB components and boards are not actually still available.

2.5.5 DSRC and 802.11p (or "WAVE")

DSRC (Dedicated Short Range Communications) is a short to medium range communications service operating at 5.9 GHz that supports both public safety and private operations in I2V and V2V communication environments. DSRC is meant to be a complement to cellular communications by providing very high data transfer rates in circumstances where minimizing latency in the communication link and isolating relatively small communication zones are important. DSRC program is intended to support a wide range of applications, of which only a small subset are presently defined. The intent is to support a wide range of I2V and V2V communications, for which most of the explicit applications cannot even be envisioned today.

The roots of DSRC may be formally traced to 2003 in the United States, when the Federal Communications Commission (FCC) adopted what is termed a Report and Order that provided licensing and service rules for DSRC in the ITS Radio Service. This enabled free, licensed use of the 5.8505-5.925 GHz frequency range, primarily for use in

safety but also for other transportation and commerce applications. It was originally conceived as a general purpose Radio Frequency Identification (RFID) technology. In making implementation decisions, however, it was quickly recognized that the Orthogonal Frequency Division Multiplexing (OFDM) was the best protocol at the time, as it allows for closely spaced orthogonal subcarriers to carry separate streams of data in parallel. There existed an emergent consumer base for wireless office LAN (Local Area Network) under IEEE 802.11a standard, based on the Atheros 5141 chipset and using OFDM. Because of the similar operating frequencies of 802.11a and the DSRC spectrum granted by the FCC, the economies of scale in using chipsets already commercially available made this case compelling during the standards discussions at the time.

For highly mobile transportation applications, a major problem to be solved in the normal operation of chipsets designed for wireless office LAN is the time, measured in seconds, required for a mobile wireless station (e.g., a laptop) to associate with a wireless access point. The nature of the channel scanning and security mechanisms in the IEEE 802.11a devices necessitates this delay, which made the existing standard association protocols unsuitable for use with safety applications that require a vehicle moving at high speed to quickly exchange information with another vehicle or RSE while in range. Changes were required at both the basic channel connection level and at the level where decisions are made to accept messages and route to their destination in order to solve this problem. There emerged three sets of standards development and associated application ideas. First, an amendment was required to the IEEE 802.11 Wireless LAN standard, to define the DSRC spectrum and band plan and to allow fast association, comprising the physical layer IEEE 802.11p. The changes at the IEEE 802.11 layer require providing new security and channel selection services at a higher layer and specifically in the control channel and multiple service channels. These are enabled, in addition to other network services, in the IEEE 1609 standards for *Wireless Access in Vehicular Environments (WAVE)* (101):

1. the resource manager, or 1609.1;
2. security services and management, or 1609.2;
3. networking services, or 1609.3;

2. PRELIMINARY CONCEPTS

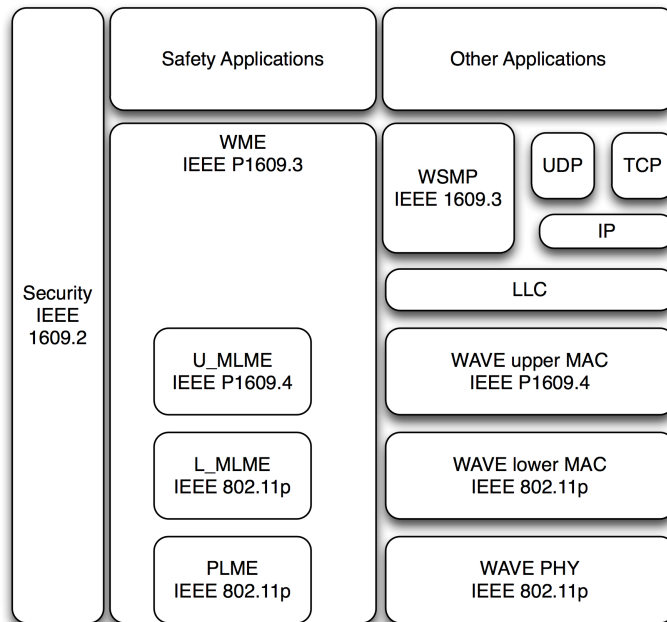


Figure 2.10: DSRC Standards DSRC standards and communication stack.

4. multichannel operations, or 1609.4.

These standards enable a system in which vehicles can connect immediately on a common channel in the DSRC band for safety applications, while other channels are available for less urgent communications. The final or top layer standard would be the application layer, (so called SAE J2735, under development by the Society of Automotive Engineers, DSRC Technical Committee).

At this point it should be valuable to spend few additional words on WAVE. The standard IEEE 802.11p (as we have seen, also referred to as WAVE) denotes enhancements to IEEE 802.11 required to support intelligent transportation systems (ITS) applications. This includes data exchange between high-speed vehicles and between these vehicles and the roadside infrastructure in the licensed ITS band of 5.9 GHz. The communications provided by WAVE generally occur over distances up to 1000 m between roadside stations and mostly high speed, but occasionally stopped and slow moving, vehicles or between high-speed vehicles. WAVE includes a number of new classes of

applications that pertain to roadway safety (e.g. vehicle collision avoidance) and emergency services (e.g. those provided by police, fire departments, ambulances and rescue vehicles) which impact in several ways: the data communications overall reliability and the extremely low latencies required. *Some critical applications require that the total time from first signal detection to completion of multiple frames of data exchange must be completed within the order of 100 ms.* In this respect, WAVE adds the capability to simplify the operations and associated management in order to support fast access.

WAVE operations utilize a control channel and multiple service channels. Prioritized access in WAVE operations uses the EDCA¹ mechanism.

The extension of the PHY for WAVE builds on the OFDM system. The WAVE radio frequency system occupies a licensed ITS radio services band, as regulated in the United States by the Code of Federal Regulations. Other regions and countries may allocate other bands in the 56 GHz range. The OFDM system provides WAVE with data payload communication capabilities of 3, 4.5, 6, 9, 12, 18, 24, and 27 Mbps in 10 MHz channels. The support of transmitting and receiving at data rates of 3, 6 and 12 Mbps is mandatory. WAVE has the option to operate on 20 MHz channels. If using the optional 20 MHz channel implementation, data payload capabilities of 6, 9, 12, 18, 24, 36, 48 and 54 Mbps can be supported. The support of transmitting and receiving at data rates of 6, 12 and 24 Mbps is mandatory for the optional 20 MHz configuration. In the context of this standard, WAVE refers to operation within the ITS band and not to operation in other bands.

Stations operating in WAVE shall be capable of transferring messages between the roadside and vehicles travelling at speeds up to 140 km/h with a packet error rate (PER) of less than 10% for PSDU lengths of 1000 bytes and between the roadside and vehicles at speeds up to a minimum of 200 km/h with a PER of less than 10% for PSDU lengths of 64 bytes. For vehicle-to-vehicle communications stations shall be capable of transferring messages at closing speeds of up to a minimum of 283 km/h with a PER of less than 10% for PSDU lengths of 200 bytes. Multipath and the effects of motion are addressed. These enabling standards hold significant promises for roadway operations and safety, thus capturing the interest of the automotive industry.

¹The term EDCA (which stands for "Enhanced Distributed Channel Access") is used in wireless networks supporting the IEEE 802.11e Quality of Service standard. It supports differentiated and distributed access to the Wireless Medium using eight different user priority sub fields supporting four different Access Categories. These are Voice, Video, Best Effort and Background.

2. PRELIMINARY CONCEPTS

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

Figure 2.11: Wireless Technologies for V2V and V2I communications. With the rapid development of information technologies, there are a number of wireless technologies which are potential for wireless InV, V2V and V2I communications.

2.5.6 Frequency and channel allocation

In Europe, ETSI (103) ¹ has presented requirements for European-wide harmonization of spectrum to CEPT (European Conference of Postal and Telecommunications Administrations) and European telecommunication administrations for deployment of ITS within the 5.9 GHz band. The frequency band 5875 to 5925 MHz has been requested for deployment of safety-related ITS applications, which require protection against interference from other services and the frequency band. The frequency band 5855 to 5875 MHz has been requested for nonsafety-related ITS, which can be operated on a nonprotected/non-interference basis.

A recent paper provided a comprehensive overview of existing approaches for the usage of the 30 MHz frequency band dedicated for safety-related car-to-car and car-to-infrastructure communication (53). Advantages and disadvantages of these approaches were analyzed based on an extensive set of evaluation criteria. These criteria are summarized as follows:

¹ *The European Telecommunications Standards Institute (ETSI)* is an independent, non-profit, standardization organization in the telecommunications industry (equipment makers and network operators) officially created in 1988 in Europe, with worldwide projection. ETSI has been successful in standardizing the GSM cell phone system and the TETRA professional mobile radio system. ETSI is officially responsible for standardization of Information and Communication Technologies (ICT) within Europe. These technologies include telecommunications, broadcasting and related areas such as intelligent transportation and medical electronics. ETSI has 740 members from 62 countries/provinces inside and outside Europe, including manufacturers, network operators, administrations, service providers, research bodies and users in fact, all the key players in the ICT arena.

- *Usability*

This represents the main requirement for safety-related car-to-car and car-to-infrastructure communication: low latency and high reliability for critical safety messages. The analysis focuses on latency, because network and/or applications should be responsible for reliability.

- *Robustness*

The wireless links robustness can be evaluated in two aspects:

1. it has to be robust in terms of bit errors (e.g., the bit error rate should be as low as possible);
2. it has to be robust in terms of interference.

- *Cost*

This criterion considers the material costs for mass production and deployment. Obviously, an inexpensive solution is preferred in order to reduce the market barrier.

- *Efficiency*

This criterion evaluates the effectiveness of channel allocation in terms of bandwidth usage. Given the scarcity of available bandwidth allocated for car-to-car and car-to-infrastructure communication, this precious resource must be used effectively.

- *Scalability*

This criterion evaluates the impact of channel allocation on the flexibility of the overall car-to-car and car-to-infrastructure communication system in different scenarios such as highways, cities, and rural areas.

- *Development effort*

This criterion considers development costs apart from material costs. A solution for channel allocation that allows a simple design and implementation of the car-to-car and car-to-infrastructure communication system is clearly preferred.

2. PRELIMINARY CONCEPTS

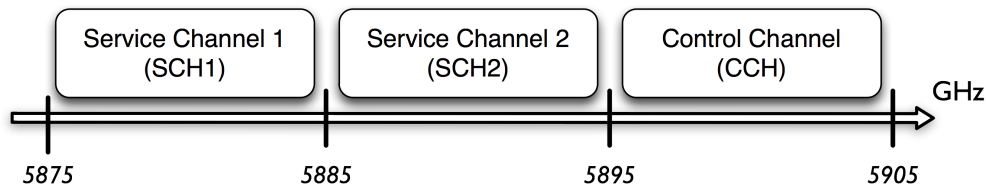


Figure 2.12: Frequency and channel allocation. 30 MHz frequency band dedicated for safety-related car-to-car and car-to-infrastructure communication where SCH1 is the first service channel, SCH2 is the secondary service channel, and CCH is the control channel.

Taking into account the basic channel usage scenarios from C2C-C Consortium¹, some scenarios are eliminated based on qualitative observations. First, currently available WLAN chipsets only support 10 and 20 MHz channels. Thus, a 30 MHz channel would require new hardware and incur significant development cost. For this reason, the usage of a 30 MHz channel is ruled out. Second, a measurement study for channel interference (78) indicates that the simultaneous usage of two adjacent channels caused significant packet loss and hence is unacceptable for ITS safety applications. Although measurements were conducted only for 10 MHz channels, similar effects are expected for simultaneous usage of 10 and 20 MHz channels. For this reason, it is not possible to divide the allocated spectrum into a 10 and 20 MHz channel. Further, because measurement results⁴⁵ show that 20 MHz channels were more susceptible to bit error rates than 10 MHz channels, the usage of 20 MHz channels is not further considered.

Although the usage of adjacent channels is possible, certain mechanisms such as WAVE (101) channel switching have to be in place to prevent simultaneous transmissions on these channels. Although interference also occurs between non-adjacent channels, it is much lower than in the case of adjacent channels. However, it is expected that packet losses caused by interference between on adjacent channels can be recovered by reliability mechanisms at the network layer and/or applications. From recent measurement

¹Car-to-Car Communication (C2C-C) Consortium is an industry consortium initiated by major European vehicle manufacturers in 2002. By 2008, the C2C-C Consortium had about 50 partners from industry and research. The main target is the creation of an open European industry standard for CAR-2-X communication based on wireless LAN technology, to support efforts to achieve interoperability, to push for harmonization of worldwide standards, and to develop deployment strategies. The major focus of the consortium is road safety and traffic efficiency applications. The C2C-C Consortium is open for car manufacturers, suppliers, research organizations, and other partners.

results (79) (91), the reasoning above and an extensive set of evaluation criteria (54), the channel allocation shown in Fig. 2.12 outperforms the best results in terms of channel allocation and interference robustness.

2.5.7 Cellular networks

Since initial analog technologies, such as the American AMPS¹ or the European TACS² cellular networks have been gradually improved in terms not only of availability all around the world, but also in the quality of service offered. As a result of applying digital communications to cellular networks, the GSM (Global System for Mobile communications) technology achieves the purpose of spreading mobile phones among normal population. Its wide adoption in Europe last years has led the expansion of GSM to other potential markets, like the Chinese one. Many people usually identify the GSM technology as the second generation (2G) of cellular networks, which substituted the first one, based on analog technologies.

Although the main concern of cellular networks, until some years ago, was focused on telephony purposes, data connections are becoming more and more popular these days. GPRS (General Packet Radio Service) appeared with the aim of providing higher data rates than the 9.6 Kbps offered by the standard GSM. GPRS provides a maximum of 177/118 Kbps in the downlink/uplink channels, and it is understood as the intermediate step between 2G and 3G, hence this is the reason why it is called 2.5G. Last years, the expansion of CDMA (Code Division Multiple Access) communication technologies

¹The "Advanced Mobile Phone System" was an analog mobile phone system standard developed by Bell Labs, and officially introduced in the Americas in 1983 and Australia in 1987. It was the primary analog mobile phone system in North America (and other locales) through the 1980s and into the 2000s. As of February 18, 2008, carriers in the United States were no longer required to support AMPS and companies such as AT&T and Verizon have discontinued this service permanently. AMPS was discontinued in Australia in September 2000. (source www.wikipedia.org)

²"Total Access Communication System" (TACS) and ETACS (Enhanced) are mostly-obsolete variants of AMPS which were used in some European countries (including the UK & Ireland in 1983). TACS was also used in Japan under the name "Japanese Total Access Communication" (JTAC). It was also used in Hong Kong. ETACS was an extended version of TACS with more channels. Both TACS and ETACS are now obsolete in Europe, having been replaced by the GSM system. In the United Kingdom, the last ETACS service operated by Vodafone was discontinued on 31 May 2001, after sixteen years of service. ETACS is however still in use in a handful of countries elsewhere in the world. NMT is another analog cellular standard that was widely used in Europe, mainly in the Nordic countries, which has now been fully replaced by GSM except for limited use in rural areas due to its superior range. (source: www.wikipedia.org)

2. PRELIMINARY CONCEPTS

has lead to the appearance of the 3G cellular networks.

CDMA2000 and UMTS (Universal Mobile Telecommunications System), this one as the evolution of GSM 2G, are two of the most extended 3G technologies. UMTS offer 384/128 Kpbs, but the recent HSPA (High Speed Packet Access) improvements offer maximum data rates of 14.4/11.5 Mbps. The introduction of cellular networks in the vehicular domain comes from several years ago, when GSM or GPRS data connections started to be used in tracking and monitoring systems. The appearance of GPRS also made possible the usage of cellular networks for providing traffic information or emergency warnings (65). However, until the arrival of 3G technologies, low data rates had avoided the spread of cellular networks in ITS (11). The advantages of the UMTS communication medium in mobility environment is defended by some authors, which use the UMTS aerial interface for direct V2V communications. The usage of the UMTS operators infrastructure in bidirectional communications is present in the literature, as monitoring systems (38) for example, but its application for V2V communications is still a challenge, due to inherent delay problems. Another drawback of using data connections with cellular networks is the extra money which has to be paid for the usage of the operators infrastructure. Current trend is paying a fixed quote per month, with an extra cost if the transmission rates fall out of the contract, but it is expected that the adoption of UMTS among the population and the vehicular field decrease the price of the final bill, by means of special agreements with operators. Apart of this, some people think that a general communication technology for the ITS domain is still needed, and cellular networks could be the solution (48).

2.5.8 Radio Data System and Traffic Message Channel

The Radio Data System (or RDS) was developed to carry digital data using the common FM radio band. This allows to multiplex additional information with the audio emission, such as the name of the radio station or the current song, but also it can include a data flag which indicates the receiver it has to pay attention to the broadcasting information because it is being transmitted a traffic bulletin. RDS offers a data rate of 1187.5 bps, and the transmission range offered by FM can reach locations at 80 kilometers far way. The RDS version deployed in U.S. is called RBDS (Radio Broadcast Data System) and operates almost identically as RDS, however its usage is less common. A more suitable solution for traffic information dissemination is offered, however,

2.6 Overview of Communication systems in VANETs

by the Traffic Message Channel (TMC) system. With this system, information about traffic problems is broadcasted digitally, so an appropriate navigation device can warn the user and calculate an alternative route, for instance. The notifications reported by TMC include an event identifier and the location of the problem. TMC traffic is usually transmitted through RDS, and this is the reason why both technologies are usually put together.

2.5.9 Satellite

Satellite communication consists of three main entities: sender station, satellite system, and receiver devices. First of all, data is sent from the sender station to the satellite, which is in charge of forwarding the information to receiver devices. Satellite communications offer a very wide coverage and a great broadcast capabilities. It is suited to provide connectivity at remote places, such as mountain areas or islands, but also in developing countries. The data can be sent from an only sender to multiple receivers at the same time and using the same frequency. Thus, satellite communications are suitable for multimedia broadcasting, such as live video, movies and music.

Although sender stations and receiver devices are usually installed at fixed locations, the later ones can be mobile and equipped in vehicles. This kind of architecture is feasible for a unidirectional system providing an I2V service, however it must be taken into account the important delay which suffer data packets, due to the propagation distance to and from the satellites. The bandwidth obtained in a mobile terminal is between 300 and 500 kbps. A sender station is usually too big to be brought inside a vehicle, and it requires a precise orientation to the satellite used. The UniDirectional Link Routing (UDLR) (95) has been standardized to emulate bidirectional communications with a satellite unidirectional link, where mobile terminals receive data using the satellite channel and transmit using other access technologies.

2.6 Overview of Communication systems in VANETs

This section presents the CAR-2-X (with this term it is intended *car-to-car* and *car-to-infrastructure* communication) system architecture and details of Geocas protocol¹

¹Geocast refers to the delivery of information to a group of destinations in a network identified by their geographical locations. It is a specialized form of multicast used by some routing protocols for

2. PRELIMINARY CONCEPTS

that serve as a basic building block for CAR-2-X communication in many European R&D projects. As shown in Figure 2.14, the CAR-2-X communication system consists of three domains: the *in-vehicle domain*, the *ad hoc domain*, and the *infrastructure domain*. The in-vehicle domain refers to a network logically composed of an *On-Board Unit* (OBU, which is responsible for CAR-2-X communication) and (potentially multiple) *Application Units* (AUs). It also provides communication services to AUs and forwards data on behalf of other OBUs in the ad hoc domain. An OBU is equipped with at least a single network device for short-range wireless communications based on IEEE 802.11p radio technology, and may also be equipped with more network devices, for example, for nonsafety communications, based on other radio technologies such as IEEE 802.11a/b/g/n (see 2.5.1 for details). An AU is typically a dedicated device that executes a single or a set of applications and uses the OBUs communication capabilities. An AU can be an integrated part of a vehicle and be permanently connected to an OBU. It can also be a portable device such as laptop, PDA, or game pad that can dynamically attach to (and detach from) an OBU. An AU and an OBU are usually connected with a wired connection, but the connection can also be wireless, using Bluetooth, WUSB, or UWB¹, for example. The distinction between AU and OBU is logical: they can also reside in a single physical unit.

The ad hoc domain, or vehicular ad hoc network (VANET), is composed of vehicles equipped with OBUs and stationary units along the road, termed road-side units (RSUs). OBUs form a mobile ad hoc network (MANET), which allows communications among nodes in a fully distributed manner without the need for centralized coordination. OBUs directly communicate if wireless connectivity exists among them.

mobile ad hoc networks. Geocast routing protocols are extremely important in vehicular networks.

¹Wireless USB is a short-range, high-bandwidth wireless radio communication protocol created by the Wireless USB Promoter Group. Wireless USB is sometimes abbreviated as "WUSB", although the USB Implementers Forum discourages this practice and instead prefers to call the technology "Certified Wireless USB" to differentiate it from competitors. Wireless USB is based on the WiMedia Alliance's Ultra-WideBand (UWB) common radio platform, which is capable of sending 480 Mbit/s at distances up to 3 meters and 110 Mbit/s at up to 10 meters. It was designed to operate in the 3.1 to 10.6 GHz frequency range, although local regulatory policies may restrict the legal operating range for any given country.

Wireless USB is used in game controllers, printers, scanners, digital cameras, MP3 players, hard disks and flash drives. Kensington released a Wireless USB universal docking station in August, 2008. It is also suitable for transferring parallel video streams, while utilizing the Wireless USB over UWB bandwidth.

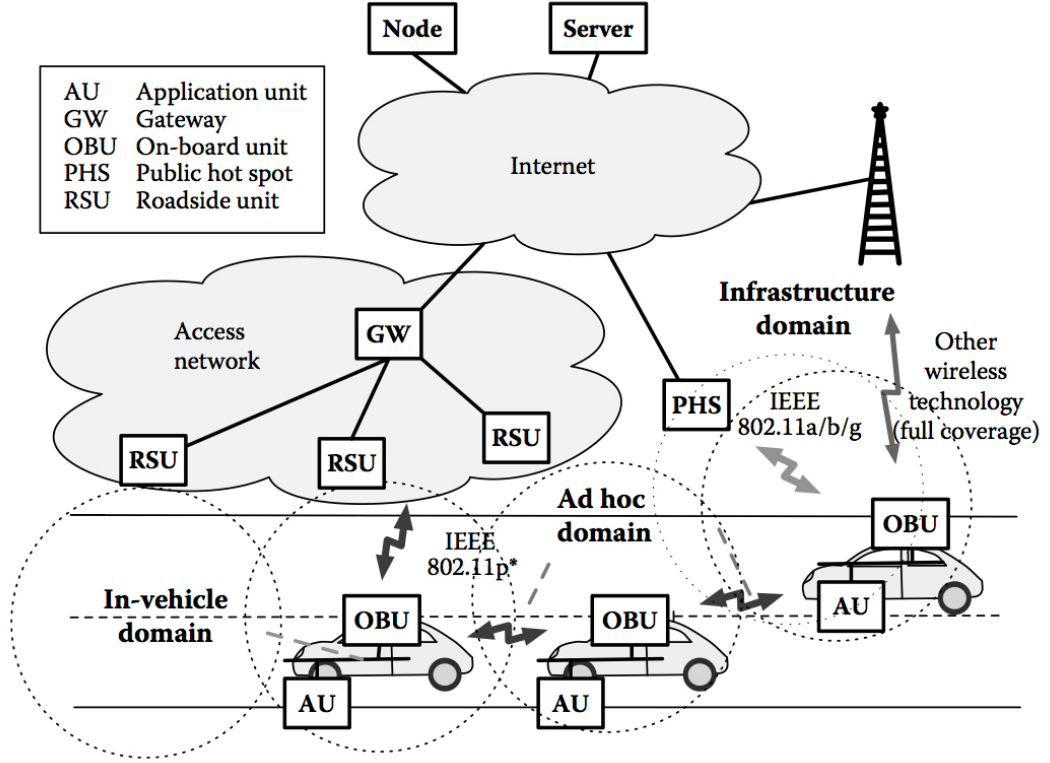


Figure 2.13: CAR-2-X communication scheme. Typical VANET scenario showing a CAR-2-X communication system and involved protocols of the IEEE 802 family. As shown in the figure, the CAR-2-X communication system consists of three domains: the *in-vehicle domain*, the *ad hoc domain*, and the *infrastructure domain*.

In the case of no direct connectivity, dedicated routing protocols allow multihop communications, where data are forwarded from one OBU to another, until they reach the destination. An RSU can be attached to an infrastructure network, which in turn can be connected to the Internet. As a result, RSUs may allow OBUs to access the infrastructure. In this way it is possible for AUs registered with an OBU to communicate with any host on the Internet, when at least one infrastructure-connected RSU is available.

2. PRELIMINARY CONCEPTS

2.7 The importance of Geocast protocols in VANETs

Geocast is basically an ad hoc routing protocol using geographical positions for data transfer. Its basic principles were originally proposed as an alternative to pure topology-based internetworking (57) and in mobile ad hoc networks (47).

Geocast assumes that vehicles acquire information about their position (i.e., geodetic coordinates) via GPS or any other positioning system. Every vehicle periodically advertises this information to its neighbor vehicles and a vehicle is thus informed about all other vehicles located within its direct communication range. If a vehicle intends to send data to a known target geographic location, it chooses another vehicle as a message relay, which is located in the direction towards the target position. The same procedure is executed by every vehicle on the multihop path until the destination is reached. This approach does not require establishment and maintenance of routes. Instead, packets are forwarded on the fly based on the most recent geographic positions. In detail, Geocast assumes that every node knows its geographical position and maintains a location table containing other nodes and their geographical positions. Geocast supports point-to-point and point-to-multipoint communication. The latter case can be regarded as group communication, where the endpoints are inside a geographical region.

Core protocol components of Geocast are *beaconing*, a *location service*, and *forwarding*. With beaconing, nodes periodically broadcast short packets with their ID, current geographical position, speed, and heading. On reception of a beacon, a node stores the information in its location table. The location service resolves a node's ID to its current position. When a node needs to know another node's position that is currently not available in its location table, it issues a location query message with the sought node ID, sequence number, and hop limit. Neighboring nodes rebroadcast this message until it reaches the sought node (or the hop limit). If the request is not a duplicate, the sought node answers with a location reply message carrying its current position and a time stamp. On reception of the location reply, the originating node updates its location table. Forwarding basically means relaying a packet towards the destination. The most innovative method for distribution of information enabled by geographical routing is to target data packets to certain geographical areas. In practice, a vehicle can

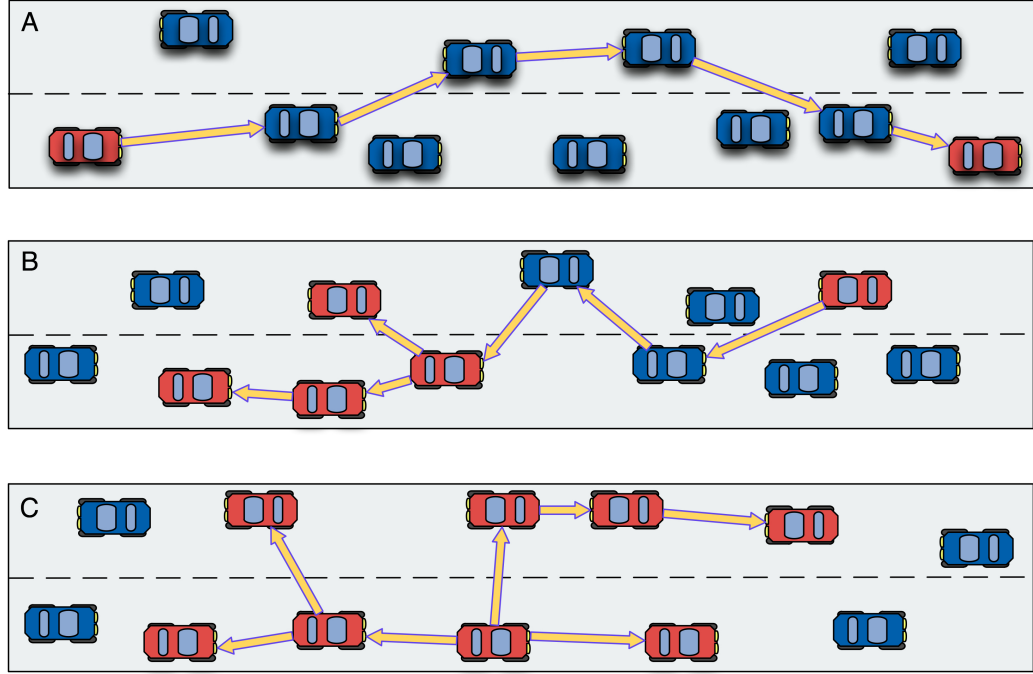


Figure 2.14: Geocasting forwarding schemes. Typical geocasting forwarding schemes: *unicast* (A), *broadcast* (B) and *topologically scoped broadcast* (C).

select and specify a well delimited geographic area to which the messages should be delivered. Once again, intermediate vehicles serve as message relays and only the vehicles located within the target area process the message and further send it to corresponding applications. In this way, only vehicles that are actually affected by a dangerous situation or a traffic notification are notified, whereas vehicles unaffected by the event are not targeted.

In summary, Geocast comprises the following forwarding schemes:

- *GeoUnicast*

According to this scheme, when a node wishes to send a unicast packet, it first determines the destinations position (by location table look-up or the location service) and then forwards the data packet to the node towards the destination, which in turn re forwards the packet along the path until the packet reaches the destination.

2. PRELIMINARY CONCEPTS

- *GeoBroadcast*

In this scheme, data packets are distributed by flooding where nodes rebroadcast the packets if they are located in the geographical region determined by the packets. This simple flooding scheme is enhanced with techniques based on packet numbering to alleviate the effects of so-called broadcast storms. Broadcast storms (a typical problem in wireless ad hoc networks) occur when multiple nodes simultaneously rebroadcast a data packet that they have just received. GeoAny-cast is similar to GeoBroadcast but addresses a single node (i.e., any node) in a geographical area.

- *Topologically scoped broadcast (TSB)*

According to this scheme, data packets are broadcasted from a source to all nodes in the n-hop neighborhood. Single-hop broadcast are a specific case of TSB and are used to send periodic messages (beacons or heartbeats).

Chapter 3

Mobility models in vehicular networks

Software development for vehicular scenarios is a very complex process because of the many factors that can impact on the overall result, ranging from the mobility of the nodes to the radio transmission and the end-to-end delay. In order to overcome or at least reduce the probability of failures in terms of application functionalities, simulation becomes a very important and mandatory step in software design before any implementation. Moreover, simulations are fast, cheap, repeatable and make it possible to investigate the influence of single parameter variations. A large number of network nodes can be simulated which is not feasible in real-world experiments. In case of new protocols' design, it is imperative to use a mobility model that accurately represents the mobile nodes (MNs) that will eventually utilize the given protocol. Only in this type of scenario it is possible to determine whether or not the proposed protocol will be useful once implemented. The faithfulness of the simulation results is proportional to the realism of the parameters and the accuracy of the models used in the simulation, in particular, the mobility model (MM) which defines the movements of the mobile nodes within the simulated area during the simulation. Obviously, the rules that describe the movements can vary according to model we want to simulate. The earlier mobility models used to simulate MANET(115) (93) assume the terrain with no obstacles and nodes to be able to move freely in the whole rectangular simulation area. This is realistic for some applications of pedestrians but inappropriate for VANET, where it is important to consider constrained routes and obstacles. Currently, there are two types

3. MOBILITY MODELS IN VEHICULAR NETWORKS

of mobility models used in the simulation of networks: *traces* and *synthetic models*(60). Traces are those mobility patterns that are observed in real life systems and provide accurate information, especially when they involve a large number of participants and an appropriately long observation period. However, new network environments (e.g. ad hoc networks) are not easily modeled if traces have not yet been created. In this type of situation it is necessary to use synthetic models. Synthetic models, instead, attempt to realistically represent the behaviors of MNs without the use of traces.

3.1 MANET'S MOBILITY MODELS

3.1.1 Entity models

In this type of models, the movement of each node is defined separately and independently from the others. Each node moves by itself, following its own parameters.

The first model of this category was described by Albert Einstein in 1926(60) in order to mimic the erratic movement that many entities show in nature by moving in extremely unpredictable ways and it is called *Random Walk Mobility Model (RWMM)*. In this mobility model, that is widely used and sometimes referred to as *brownian motion*, a MN moves from its current location to a new location by randomly choosing a direction and speed in which to travel. The new speed and direction are both chosen from predefined ranges, $[speedmin, speedmax]$ and $[0, 2\pi]$ respectively. Each movement in the Random Walk Mobility Model occurs in either a constant time interval t or a constant distance traveled d , at the end of which a new direction and speed are calculated. If a MN, which moves according to this model, reaches a simulation boundary, it bounces off the simulation border with an angle determined by the incoming direction. The MN then continues along this new path. Many derivatives of the Random Walk Mobility Model have been developed including the 1-D, 2-D, 3-D, and d-D walks. In 1921, Polya proved that a random walk on a one or two-dimensional surface returns to the origin with complete certainty, i.e., a probability of 1.0(116). This characteristic ensures that the random walk represents a mobility model that tests the movements of entities around their starting points, without worry of the entities wandering away never to return. One of the main issues of RWMM is that it is a memoryless mobility pattern because it retains no knowledge concerning its past locations and speed values, thus generating unrealistic movements such as sudden stops and sharp turns.

The most popular random mobility model is the *Random Waypoint (RWP)*(44), largely used for simulating ad hoc networks and available in many simulators like *ns2*(3), *Glo-MoSim*(2) and *Qualnet*(4). According to this model, a MN stays in a location and waits for a certain amount of time (i.e. *pause time*). Once this time expires, the MN chooses a new random location within the simulation area and then travels towards it at a selected speed, uniformly distributed in a predefined interval. Usually, MNs are initially distributed randomly and uniformly in the simulation area but it is important to realize that this is not representative of the manner in which nodes distribute themselves when moving. One of the most important parameter in RWP simulations is the average MN neighbor percentage, given as the cumulative percentage of a total MNs that are a given MN's neighbor. In fig. 3.4 50 MNs move accordingly to RWP for roughly half an hour; it is easy to realize that there is high variability during the first 500 seconds and this effect should be considered in short simulations, as they can seriously affect the performance results. Another problem that usually affects RWP is the so called "*density waves*", where a density wave is the clustering of nodes in a particular region of the simulated area (in RWP this occurs frequently in the center of the map). To overcome this problem and promote a semi-constant number of neighbors throughout the simulation, the *Random Direction Mobility Model (RDMM)* was developed. In this model, MNs choose a random direction and then travel to the border of the simulation area in that direction; once the area boundary is reached, like the RWP model, MNs pause for a specified time, choose a direction - in the range between 0 and 180 degrees - and continue the process. A slight modification to RDMM is the *Modified Direction Mobility Model (MDMM)*, where MNs can change direction without reaching the boundary of the simulation area.

Although the Random Waypoint based mobility models are widely accepted for their simplicity of implementation and analysis, there are some problems related to:

- *temporal dependency*: due to physical constraints of the mobile entity itself, the velocity of mobile node will change continuously and gently instead of abruptly, i.e. the current velocity is dependent on the previous velocity. However, the velocities at two different time slots are independent in the RWP based models;
- *spatial dependency*: the movement pattern of a mobile node may be influenced by and correlated with nodes in its neighborhood. In RWP based models, instead,

3. MOBILITY MODELS IN VEHICULAR NETWORKS

each mobile node moves independently of others;

- *geographic restrictions*: in many cases, the movement of a mobile node may be restricted along the street or a freeway. A geographic map may define these boundaries.

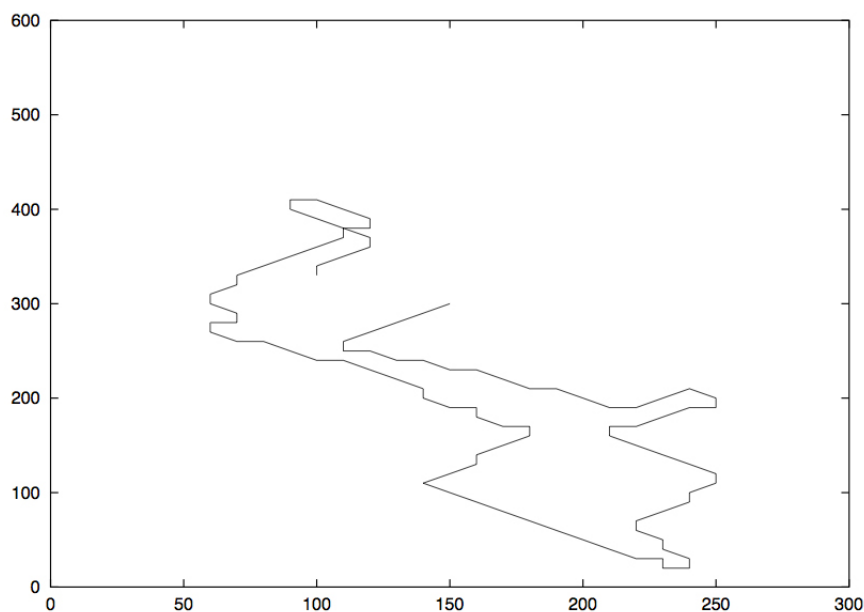


Figure 3.1: Random Walk Mobility model. Traveling pattern of a MN using a probabilistic version of RWM.

Two other interesting mobility models (which however are outside of the scope of this work) are the *Boundless Simulation Area*, where MNs that reach one side of the simulation area continue traveling and reappear on the opposite side of the simulation area, and the *Gauss-Markov model*, designed to adapt to different levels of randomness using one tuning parameter. As a conclusion to the Entity Models, we can say that RWMM, by producing a Brownian motion, evaluates a static network when used to a performance investigation. With a large input parameter - that can be distance or time - is similar to the RWP model without pause times. The main difference between these two mobility models is the density wave in the center of the simulation area observed in RWP. On the other side, RWP model is used in many prominent simulation studies of ad hoc network protocols because it is flexible and it appears to create "quite" realistic

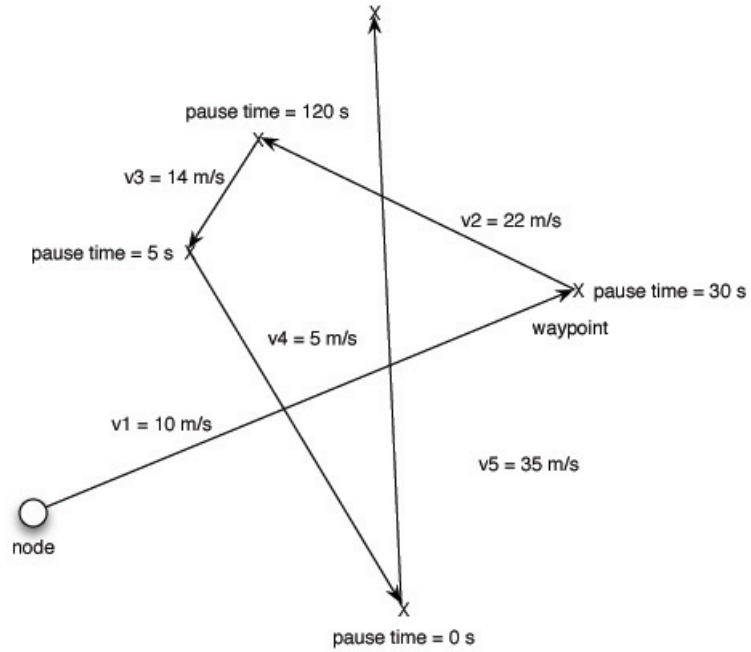


Figure 3.2: Random-Wapoint model In RWP, mobile nodes choose randomly a destination and move towards it with a constant velocity.

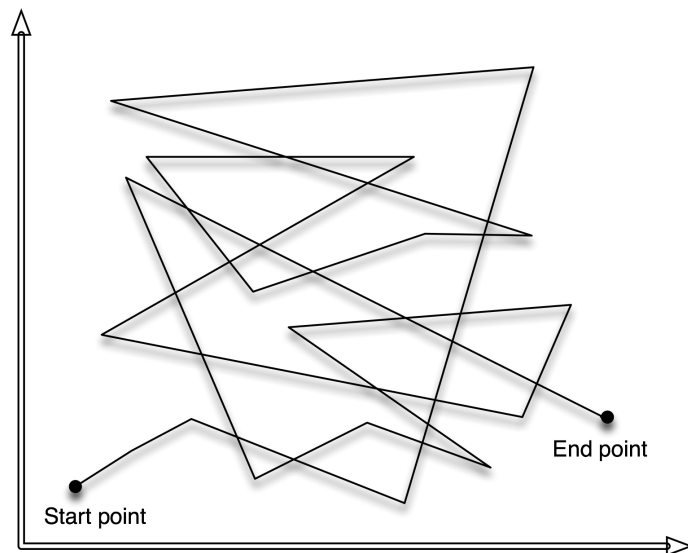


Figure 3.3: Random-Wapoint trace. Typical RWP's motion trace in a 500x500 area.

3. MOBILITY MODELS IN VEHICULAR NETWORKS

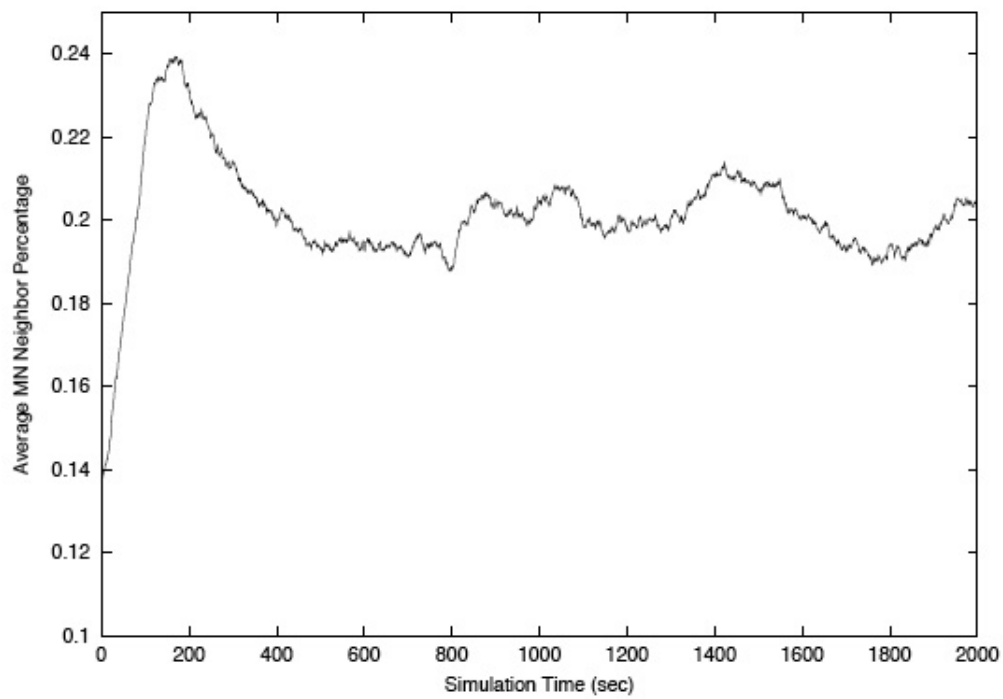


Figure 3.4: Random-Waypoint neighbors percentage. Average neighbors percentage vs. time.

mobility patterns for the way people might move in, for example, a conference setting or a museum. One concern with this model is the straight movement pattern followed by the MNs to reach the chosen destination. We will see, later on, that this model is not suitable to simulate VANET scenarios. Regarding to RDMM, we can say this model is realistic because it is unlikely that people would spread themselves evenly throughout an area (a building or a city). In addition, it is also unlikely that people will only pause at the edge of a given area. The MDMM, instead, allows MNs to pause and change directions before reaching the simulation boundary but it is somehow similar to the RWMM. The Boundless Simulation Area mobility model provides movements that can be expected in the real world and this is the only model that allows MNs to travel unobstructed in the simulation area, thus removing any simulation edge effects from the performance evaluation. One concern, however, is the undesired side effects that would occur when MNs can become neighbors again and again during the simulation time. This problem has to be taken into consideration when you want to simulate and evaluate performances of data dissemination protocols. As it concerns the Gauss-Markov mobility model, we can say that it provides movement patterns that one might expect in a real world (if appropriate parameter are chosen) but it prevents MNs to go close to the simulation area's edges.

3.1.2 Group models

This class of mobility models represents movements of groups of nodes whose actions are completely independent of each other and describes very well many scenarios such as in tourist trips, where groups of tourists move together to visit particular monuments or in battlefields, where group of soldiers in a military scenario may be assigned the task of searching a particular plot of land in order to destroy land mines, capture enemy attackers, or simply work together in a cooperative manner to accomplish a common goal. Moreover, group models can easily and effective depict scenarios such as avalanche rescue, where the responding team consisting of human and canine members work cooperatively. The human guides tend to set a general path for the dogs to follow, since they usually know the approximate location of victims. The dogs each create their own random paths around the general area chosen by their human counterparts. Another example that can be represented by this model is a visit of a group of students to a museum. The museum guide who moves in the different parts of the museum

3. MOBILITY MODELS IN VEHICULAR NETWORKS

represents the reference point: the students follow him during the visit, and may make some movements individually around him.

The most general model of this group is the *Reference Point Group Mobility Model (RPGM)*(39), which represents a random motion of a group of nodes as well as a random motion of each individual within the group. Group movements are based upon the path traveled by a logical center for the group, which is used to calculate group motion via a group motion vector, $\vec{GM}$ (see figure 3.5). The motion of the group center completely characterizes the movement of its corresponding group of MNs, including their direction and speed. Individual MNs randomly move about their own predefined reference points, whose movements depend on the group movement. As the individual reference points move from time t to $t + 1$, their locations are updated according to the groups logical center. Once the updated reference points, $RP(t + 1)$, are calculated, they are combined with a random motion vector, $\vec{RM}$, to represent the random motion of each MN about its individual reference point. Figure 3.5 gives an illustration of three MNs moving with the RPGM model. The figure illustrates that, at time t , three black dots exist to represent the reference points, $RP(t)$, for the three MNs. As shown, the RPGM model uses a group motion vector $\vec{GM}$ to calculate each MNs new reference point, $RP(t + 1)$, at time $t + 1$; as stated, $\vec{GM}$ may be randomly chosen or predened. The new position for each MN is then calculated by summing a random motion vector, $\vec{RM}$, with the new reference point. The length of $\vec{RM}$ is uniformly distributed within a specied radius centered at $RP(t+1)$ and its direction is uniformly distributed between 0 and 2π . Movement patterns using the RPGM model are shown in figure 3.6. Figure 3.7 is an illustration of ve groups moving, such that each group has a different number of MNs. Both the movement of the logical center for each group, and the random motion of each individual MN within the group, are implemented via the Random Waypoint Mobility Model. One difference, however, is that individual MNs do not use pause times while the group is moving. Pause times are only used when the group reference point reaches a destination and all group nodes pause for the same period of time.

3.2 VANET'S MOBILITY MODELS

This section gives an overview of the models that are specifically proposed for simulating VANETs, or used in the general MANET context but still valid in VANET. Basically,

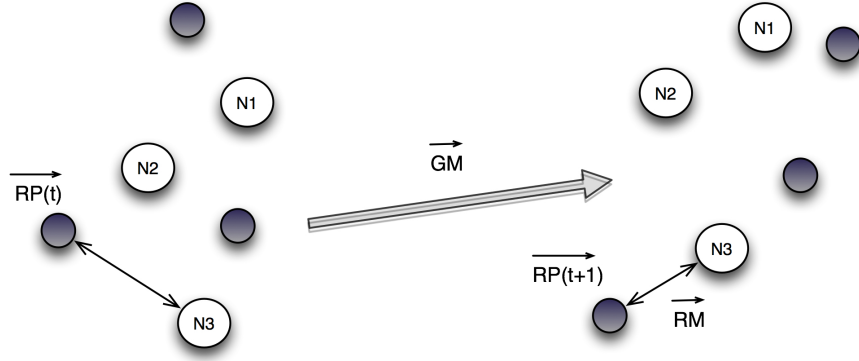


Figure 3.5: RPGM model Movements of three nodes according to the RPGM mobility model.

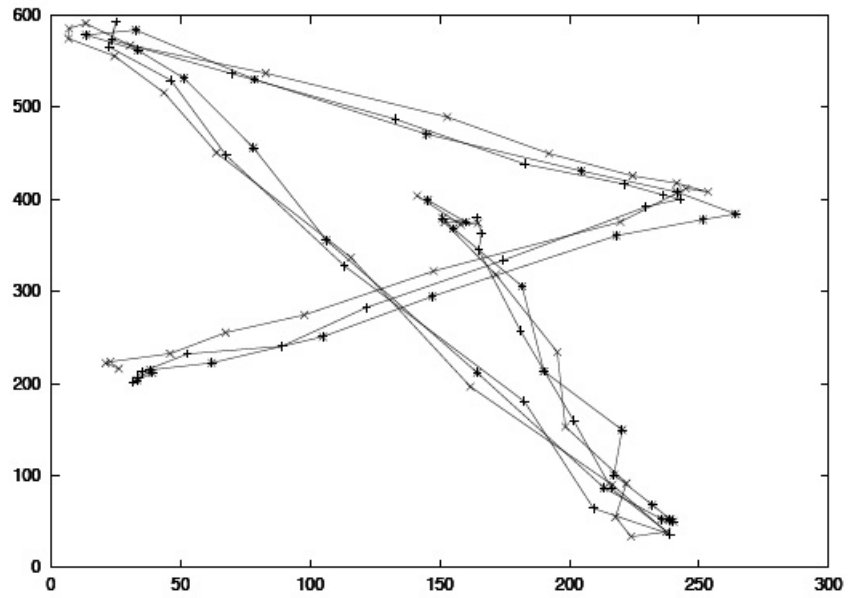


Figure 3.6: RPGM group pattern. Traveling group pattern of three nodes using the RPGM mobility model.

3. MOBILITY MODELS IN VEHICULAR NETWORKS

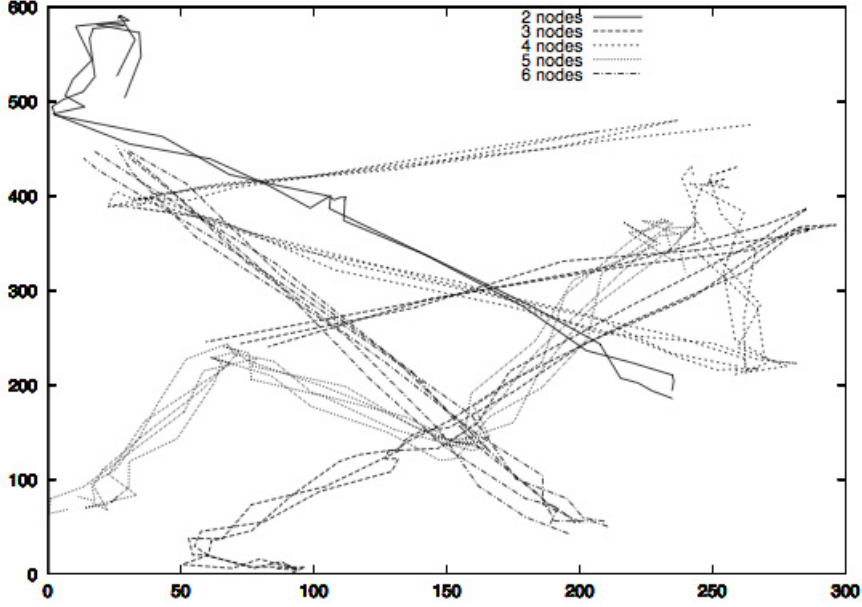


Figure 3.7: RPGM group pattern (5 mobile nodes). Traveling group pattern of five nodes using the RPGM mobility model.

these models simulate movements in routes, contrary to the previous ones that assume an open area. As we will see, the considered parameters differ from a model to another. For instance, some models use traffic control mechanisms (stop signs or traffic lights) at route intersections, and others just assume continuous movement at these points. Some assume routes to be single-lane, some others support multi-lanes routes. Some define the security distance, while others just ignore this parameter. Whereas it is crucial to test and evaluate protocol implementations in real testbeds environments, logistics difficulties, economic issues and technology limitations make simulation the mean of choice in the validation of networking protocols for VANETs and widely adopted a first step in development of real world technologies. A critical aspect in a simulation study of VANETs is the need for a mobility model which reflects, as close as possible, the real behavior of vehicular traffic.

When dealing with vehicular mobility modeling, it is worth to introduce two concepts: *macro-mobility* and *micro-mobility*. With the former, we mean all the macroscopic aspects which influence vehicular traffic, such as: road topology, constraining car movements, per-road speed limits, number of lanes, overtaking, safety rules, traffic signs

establishing the intersections' crossing rules. The latter, instead, refers to the driver's individual behavior when interacts with other drivers or road infrastructure (traveling speed in different traffic conditions, acceleration, deceleration and overtaking criteria, driver's attitude, generally related to sex, age and/or mood).

Obviously, it would be desirable for a trustworthy vanet simulation that both micro and macro mobility descriptions be jointly considered. However, many mobility models employed for VANETs ignore these guidelines, thus failing in reproducing peculiar aspects of vehicular motion such as car acceleration and deceleration in presence of nearby vehicles, queuing at road intersections clustering caused by semaphores, vehicular congestion and traffic jams.

3.2.1 Factors affecting mobility in VANETs

As said just before, mobility pattern of nodes in a VANET can significantly influence route discovery, maintenance, reconstruction, consistency and caching mechanism and this can obviously affect data dissemination protocols.¹ For example, static or slow moving vehicles tend to hold the topology configuration, as they can be assimilated to "semi-static" and reliable relaying nodes; on the contrary, fast moving vehicles can cause highly changeable topology, demanding frequent route reorganization and packet losses. In detail, the following factors can be considered in modeling VANETs' mobility:

- *block size*: if the block (that is the smallest area surrounded by streets) is extended, it causes few intersections and, then, in turn, the frequency of which the vehicles stop decreases;
- *streets layout*: streets force nodes to confine their movements to well-defined paths, determining the spatial distribution of nodes and their connectivity. Moreover, streets can have either single or multiple lanes and can allow one-way or two-way traffic;
- *average speed*: vehicle's speed determines how quickly its position changes, which in turn determines the rate of network topology change. The speed limit of each

¹In (61), authors proved that some aspects of mobility (such as multiple lanes and synchronization at traffic signals) can be ignored as they have only a marginal impact on the overall ad-hoc routing performances. On the contrary, performance indexes like *delivery ratio* and *packet delay* are more sensitive to the clustering effect of vehicles waiting at intersections and/or acceleration/deceleration.

3. MOBILITY MODELS IN VEHICULAR NETWORKS

road, as well as acceleration and deceleration, also directly affects the average speed of vehicles and how often the existing routes are broken or new routes are established;

- *interdependent vehicular motion*: in sophisticated mobility models, the movement of each vehicle is influenced by the movement pattern of its surrounding vehicles (minimum safety distance, lane changing, overtaking);
- *traffic control mechanism*: stop signs and traffic lights affect the mobility and can result in the formation of clusters and queues. Furthermore, reduced mobility implies more static nodes and then, slower rates of route changes in the network. On the contrary, vehicles' clusters can affect network performance due to channel contention and longer network partitions.

3.2.2 Freeway

This model was introduced in(25) to emulate the movement pattern of mobile nodes on streets defined by maps and includes in the simulation area many freeways, each side of which is composed of many lanes. No urban routes, thus no intersections are considered in this model and the map is composed of a number of horizontal and vertical streets which are made up, in turn, of two lanes for each direction.

At the beginning of the simulation, the mobile nodes are randomly placed in the lanes and move using history-based speeds. The speed of each vehicle follows the following formula:

$$V(t + 1) = V(t) + random() \times a(t)$$

Where $V(t)$ is the vehicle's speed at time t , $random()$ returns a random value in the interval $[-1, 1]$ and $a(t)$ is the acceleration of the vehicle at time t . A security distance should be maintained between two subsequent vehicles in a lane, so if the distance between two vehicles is less than this required minimal distance, the second one decelerates ($a(t)$ is forced to be < 0) and let the forward vehicle moves away. The change of lanes is not allowed in this model thus the vehicle moves in the lane it is placed in until it reaches the simulation area limit, then it is placed randomly in another position and repeats the process. This scenario is definitely unrealistic.

3.2.3 Manhattan

Like the freeway model, this is a generated-map-based model introduced in(25) to simulate a urban environment. The simulation area is represented by a map (generated before the simulation start) containing vertical and horizontal roads made up of two lanes, allowing the motion in the two directions (north-south for the vertical roads and east-west for the horizontal ones). Before starting the simulation, vehicles are randomly placed on the roads and then start moving continously according to history-based speeds following the same formula used in the freeway model. When reaching a crossroads, the vehicle randomly chooses a direction to follow that is continuing straightforward, turning left or turning right according to a triplet of probabilities. Contrary to the the freeway model, a vehicle can change a lane when it passes a crossroads with absolutely no control mechanism, thus continuing their movements without stopping. This makes this model unrealistic.

3.2.4 City Section Mobility

In *City Section Mobility Model*(15), the simulation area is a street network that represents a section of a city where the ad hoc network exists. The streets and speed limits are based on the type of city being simulated. For example, the streets may form a grid in the downtown area of the city with a high-speed highway near the border of the simulation area to represent a loop around the city. Each MN begins the simulation at a defined point on some street and then, randomly choose a destination, also represented by a point on some street. The movement algorithm from the current destination to the new destination locates a path corresponding to the shortest travel time between the two points; in addition, safe driving characteristics such as a speed limit and a minimum distance allowed between any two MNs exists. Upon reaching the destination, the MN pauses for a specified time and then randomly chooses another destination (i.e., a point on some street) and repeats the process.

This model can be seen as a hybrid model between RWP and Manhattan, as it introduces the principle of RWP like the pause-time and random selection destination, within a generated-map-based area. Anyway, contrary to RWP, the application domain (the union of the segments defined by the edges of the space graph) is unconvex.

3. MOBILITY MODELS IN VEHICULAR NETWORKS

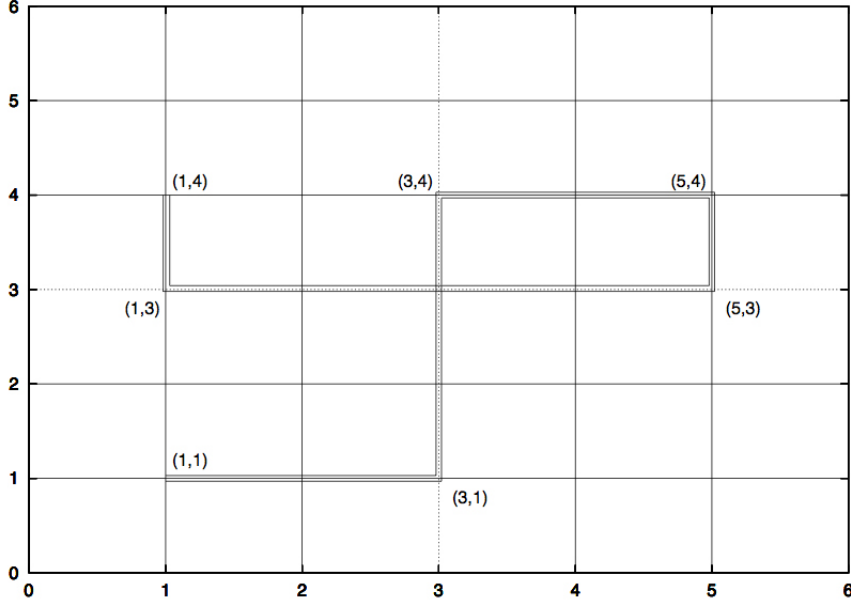


Figure 3.8: City Section Mobility Model. Traveling pattern of a MN using the CSMM.

Figure 3.8 show the movements of a MN using an example city section according to the CSMM. Within this example, the center-most vertical and horizontal streets are designated as mid-speed roads (i.e., $x = 3$ and $y = 5$), similar to main thoroughfares within a city; all other roads are considered to be slow residential roads. A MN starts the simulation at $(1, 1)$, moves to $(5, 4)$ and then moves to $(1, 4)$. The dashed lines in fig. 3.8 indicate the mid-speed roads and the double lines represent streets traveled by the MN in our example. As shown, both moves from $(1, 1)$ to $(5, 4)$ and from $(5, 4)$ to $(1, 4)$ use mid-speed roads.

The CSMM provides realistic movements for a section of a city since it severely restricts the traveling behavior of MNs. In other words, all MNs must follow predefined paths and behavior guidelines (e.g. traffic laws). In the real world, MNs do not have the ability to roam freely without regard to obstacles and traffic regulations. In addition, people typically tend to travel in similar patterns when driving across town or walking across campus. Enforcing that all MNs follow predefined paths will increase the average hop count in the simulations compared to other mobility models. Improvements of the CSMM include pause times at certain intersections and destinations, incorporate

acceleration and deceleration, and account for higher/lower concentrations of MNs depending on the time of day. In addition, the model should be expanded to include a larger simulation area, an increased number of streets, a high-speed road along the border of the simulation area, and other novel path-finding algorithms.

3.2.5 Rice University Model (RUM)

RUM(84) is very similar to CSM but, on the contrary, it uses real maps obtained from the TIGER/Lines database(15). For each route segment 1, the coordinates are extracted and converted using the Mercator projection(84). The extracted points are then presented by a graph, where the crossroads are presented by vertices, and routes by weighted arcs. The weight of each arc is dynamically calculated in such a way to mimic the estimated time required for a vehicle to move over the corresponding segment, which is proportional to its maximum authorized speed, its distance, and the number of vehicles it currently contains. Therefore, the lower the weight, more the vehicles move freely in the segment. Note that the maximum authorized speed of a route segment depends on its type. RUM has been compared to RWP, in the evaluation of DSR(44) in two different regions(84). The first one contains a lot of roads of the same type, with a maximum speed limit set to 56km/h, while the second one has fewer roads, but of different types (which ranges the maximum speed from 56km/h to 120km/h). The two models engender almost the same results in the first region simulation. However, in the second region simulation (where the mobility is more constrained) the results were clearly different. Moreover, like all the previous models, RUM denies no control mechanisms at crossroads.

3.2.6 Stop Sign Model (SSM)

SSM(77) is the first model that integrates a traffic control mechanism. In every crossroads, a stop signal is put, which obliges vehicles to slow down and make a pause there. This model is based on real maps of the TIGER/Lines database, but all roads are assigned a single lane in each direction. A vehicle should never overtake its successor (like in all the models presented before), and should tune its speed to keep the security distance. If many vehicles arrive at an intersection at the same time, they make a queue and then each one waits for its successor to traverse the crossroads. This results in gathering of nodes, and hugely affects the network connectivity as well as the vehicle

3. MOBILITY MODELS IN VEHICULAR NETWORKS

mobility (average speeds). According to the authors(77), the problem with this model is the unrealistic disposition of the stop signals, since it is impossible to find a region with stop signals at each intersection. Therefore, they proposed TSM(77), described hereafter.

3.2.7 Traffic Sign Model (TSM)

In this model, stop signals are replaced by traffic lights. A vehicle stops at a crossroads when it encounters a red stoplight, otherwise it continues its movement. When the first vehicle reaches the intersection, the light is randomly turned red with probability p (thus turned green with probability $1 - p$). If it turns red it remains so for a random delay (pause-time), forcing the vehicle to stop as well as the ones behind it. After the delay, it turns green then the nodes traverse the crossroads one after the other until the queue is empty. When the next vehicle arrives at the crossroads the process is repeated. TSM and SSM has been evaluated by simulation with ns2(3). The results show that the two models are not significantly influenced by the speed of nodes (maximum speeds). This is due to the traffic control models, which slow down the nodes and give more stability to the network(77). When increasing the pause-time at the intersections, the authors remarked that the performances improved for both models, and that SSM gives better results than TSM when using the same pause-time. The authors argue this by the fact that in SSM nodes always stop at the intersections, unlike TSM. Nevertheless, in reality the pause-time of stop signals is shorter than the one of traffic lights, which makes TSM more stable indeed(77).

3.2.8 STRAW

STRAW(21) is also a model relying on real maps of TIGER/line. Like the other models (except freeway), the roads include one lane in each direction, and is divided into segments. The model is basically composed of three modules: intra-segment mobility manager, inter-segment mobility manager, and finally the route management and execution module. At the beginning of the simulation, the nodes are placed randomly one behind the other. Then they move using the car following model(21) such that they try to accelerate until reaching the maximum speed of the segment. The first module manages this movement until reaching an intersection. The security distance is maintained, and the overtaking is not allowed. At crossroads the vehicles always slow

down, even when they change a segment and turn without a full stop, which is realistic. The second module defines the traffic control mechanism including both stop signals and traffic lights, put on crossroads according to the class of the intersected routes. In addition to this usual control form, the module makes sure that the next segment to take contains enough available space before moving the vehicle toward it. If it is fully busy, the vehicle waits at the crossroads (at the end of the first segment). The last module selects the routes to be taken by each vehicle during the simulation. It implements two approaches: simple straw and straw OD. In the first one, the direction is randomly selected at each intersection. That is, when reaching an intersection, the vehicle randomly decides whether to continue straightforward, or to turn and change the route. On the other hand, in the second approach a destination is selected toward which the vehicle moves using the shortest path. The simulation study made by the authors (21) shows that when using STRAW the reception ratio decreases from 43% up to 53% compared to movements in an open area. The results of this simulation also illustrate that the routes arrangement has an impact; scenarios with a high number of crossroads slow down the average speeds of nodes, which improves the reception ratio.

3.2.9 Gorgorin et al. model

In addition to all the realistic parameters of the previous models, this one(18) implements an overtaking mechanism within multi-lane segments. A vehicle always tries to move on the most right lane (the lowest rang), except in case of overtaking during which it moves left, and intersections in urban environment where it chooses the lane according to the next direction.

This model belongs to the *psycho-physical models*(87), which divide the bi-dimensional space of distance Δx and speed difference Δv with respect to front vehicle into several areas corresponding to different drivers behaviors. According to this, we can have the following four driving modes:

- *no reaction*: no influence is exerted from the ahead vehicle, which is too distant or traveling at higher speed than the considered vehicle. Under this circumstance, cars are free to reach and keep the desired speed;
- *reaction*: the vehicle is approaching the car ahead and, as a consequence, it has to reduce its speed to keep at safety distance;

3. MOBILITY MODELS IN VEHICULAR NETWORKS

- *unconscious reaction*: the vehicle is in short distance with respect to the car in front, the speed difference is small and the two vehicles build up a kind of *cluster*;
- *deceleration*: in this case, the distance is too small if compared to the current speed difference, and the vehicle under consideration is forced to brake.

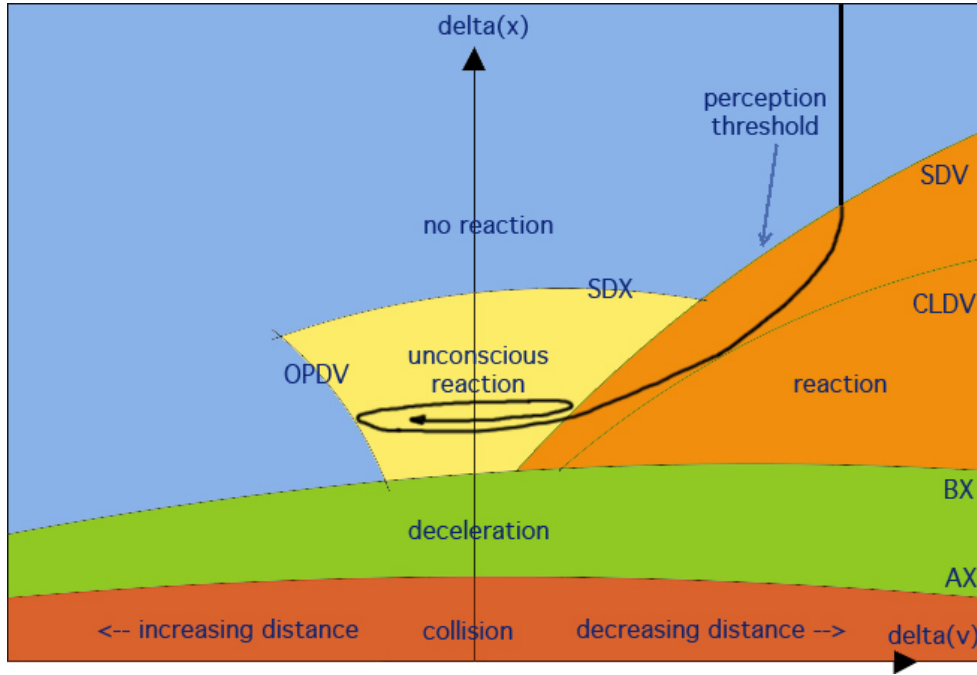


Figure 3.9: Psycho-physical vehicular interaction. The Δx axis divides positive (vehicle approaching front car) and negative (front vehicle increasing its distance) speed difference Δv . A possible trajectory is shown by the narrow: a vehicle, initially in free flow condition (i.e in the no reaction zone), starts approaching a vehicle in front. As long as the distance is high enough, the driver ignores the obstacles ahead, keeping his/her pace. However, when the distance becomes too small with respect to the current speed (i.e. the reaction zone is entered) the driver starts slowing down. The braking process brings the driver to match the front car speed ($\Delta v = 0$). Any increase in the speed if the ahead vehicle is then matched by the back vehicle with some delay, due to the finite reaction time of drivers ($\Delta v < 0$). For the same reason, any reduction in front car's speed is matched by the back vehicle with delay as well (the last segment of the movement curve, in the $\Delta v > 0$, unconscious area).

These behavioral areas are separated by thresholds, which trigger consistent acceleration changes. In the example of fig.3.9, AX and BX are the minimum and desired

safety distances, *SDV* and *CLDV* are the distances at which the driver perceives he is approaching a slower vehicles and ends the approaching procedure, *OPDV* and *SDX* model the distances beyond which the driver understands he is free to accelerate. All these distances are obviously functions of the speed difference. The main challenge of this kind of mobility models lies in the characterization of such thresholds.

Finally, note that the model includes both traffic lights and stop signals at intersections. One of these two different control mechanisms is put at each intersections according to the types of the intersecting segments. The most important parameter added in this model is the overtaking mechanism. However, no study investigating this issue has been done yet.

3.2.10 Metrics for mobility models

In this chapter is often highlighted the importance of choosing the right mobility model for a simulation of a VANET protocol, because it can heavily affect the results ¹. Different mobility models lead to different mobility patterns but models themselves do not give clear images how mobility patterns are different each other, that is why some mobility metrics are required to describe these mobility patterns. In this scenario, the definition of precise performance metrics becomes essential but it is not trivial as it can seem.

The best contribute on this problem is given by "Important", a framework thought to systematically analyze the Impact of mobility on routing protocols for ad-hoc networks(25) (intuitively, the results can be easily extended to any other application deployed in a VANET). The basic idea was simulating and then analyzing three main routing protocols, largely used in mobile networks, such as DSR, DSDV and AODV, over different mobility patterns.

Very important are the so called *protocol independent metrics* (or *direct metrics*), which are measurements with clear physics meaning used to extract the characteristics of mobility and the connectivity graph between the mobile nodes (i.e. host speed or relative

¹i.e., in 2006 V. Naumov, R. Baumann and T. Gross compared AODV and GPSR in ns-2 with Random Waypoint model and real traces collected in the city of Zurich. The results were impressive: with RWP, both the protocols reported a delivery ratio of 99%, value that decreased with real traces up to 34% and 13% respectively(69)

3. MOBILITY MODELS IN VEHICULAR NETWORKS

speed); in addition, we can find the *derived mobility metrics* (such as graph connectivity), measurements derived from physical observations through mathematical modeling. Before formally defining the metrics, it is mandatory to introduce the following terminology:

1. $\vec{V}_i(t)$: velocity vector of node i at instant t ;
2. $|\vec{V}_i(t)|$: speed of node i at instant t ;
3. $\theta_i(t)$: angle made by $\vec{V}_i(t)$ at time t with the x-axis;
4. $\vec{a}_i(t)$: acceleration vector of node i at time t ;
5. $x_i(t)$: x-coordinate of node i at instant t ;
6. $y_i(t)$: y-coordinate of node i at instant t ;
7. $D_{i,j}(t)$: euclidean distance between nodes i and j at time t ;
8. $RD(\vec{a}(t), \vec{b}(t'))$: relative direction (or cosine of the angle) between the two vectors;
9. $SR(\vec{a}(t), \vec{b}(t'))$: speed ratio between the two vectors;
10. R : transmission range of a mobile node;
11. N : number of mobile nodes;
12. T : simulation time;
13. $random()$: returns a value uniformly distributed in the interval $[-1, 1]$.

Given the terminology, we can describe the metrics used to differentiate mobility patterns starting from *direct metrics*:

- *degree of spatial dependence*

This metric can be defined as the extent of similarity of the velocities of two nodes that are not too far apart. Formally, we have:

$$D_{spatial}(i, j, t) = RD(\vec{v}_i(t), \vec{v}_j(t)) * SR(\vec{v}_i(t), \vec{v}_j(t)) \quad (3.1)$$

The value is higher when the two vehicles travel at almost the same speed and direction. However, it decreases if relative direction or speed ratio decreases. It is possible to define also the *average degree of spatial dependence* as the $\bar{D}_{spatial}(i, j, t)$ averaged over node pairs and time instants satisfying certain condition. Its value is small when vehicles move independently, while it increases when the node's movement is somehow coordinated and they have almost the same directions and speeds:

$$\bar{D}_{spatial} = \frac{\sum_{t=1}^T \sum_{i=1}^N \sum_{j=i+1}^N D_{spatial}(i, j, t)}{P} \quad (3.2)$$

where P is the number of tuples (i, j, t) such that $D_{spatial}(i, j, t) \neq 0$.

- *degree of temporal dependence*

This metric can be defined as the extent of similarity of the velocities of a node at two time slots that are not too far apart. Formally, we have:

$$D_{temporal}(i, t, t') = RD(\vec{v}_i(t), \vec{v}_i(t')) * SR(\vec{v}_i(t), \vec{v}_i(t')) \quad (3.3)$$

The value is high when the vehicle travels at almost the same speed and direction over a certain interval of time that can be defined; it decreases, instead, if relative direction or speed ratio decreases. It is possible to define also the *average degree of temporal dependence* as the $\bar{D}_{temporal}(i, t, t')$ averaged over node pairs and time instants satisfying certain condition. This value is small when the node is completely independent of its velocity, while it increases when the node's movement is strongly dependent on the velocity at some previous step:

$$\bar{D}_{temporal} = \frac{\sum_{i=1}^N \sum_{t=1}^T \sum_{t'=1}^T D_{temporal}(i, t, t')}{P} \quad (3.4)$$

where P is the number of tuples (i, t, t') such that $D_{temporal}(i, t, t') \neq 0$.

- *relative speed (RS) and average speed (AS)*

This metric can be defined with the standard definition from physics:

$$RS(i, j, t) = \left| \vec{V}_i(t) - \vec{V}_j(t) \right| \quad (3.5)$$

and in case of $D_{spatial}(i, j, t)$, it is possible to add the following condition:

$$D_{i,j}(t) > k * R \Rightarrow RS(i, j, t) = 0 \quad (3.6)$$

3. MOBILITY MODELS IN VEHICULAR NETWORKS

where k is a constant that can be determined through simulations.¹

At the same time, it is possible to define the *average relative speed* as the value $RS(i, j, t)$ averaged over node pairs and time instants satisfying certain condition. Formally:

$$\bar{RS} = \frac{\sum_{i=1}^N \sum_{j=1}^N \sum_{t=1}^T RS(i, j, t)}{P} \quad (3.7)$$

where P is the number of tuples (i, j, t) such that $RS(i, j, t) \neq 0$.

Another way to define this metric can be the following: if $P(m, t)$ and $P(n, t)$ are the positions of hosts m and n at time t , respectively, the *relative velocity* between m and n can be defined as follows:

$$V(m, n, t) = \frac{d(P(m, t) - P(n, t))}{dt} \quad (3.8)$$

Averaging the absolute value of relative velocity over time, we have:

$$M_{m,n} = \frac{1}{T} \int_{t_0}^{t_0+T} |V(m, n, t)| dt \quad (3.9)$$

that is the *average relative speed* $M_{m,n}$ of vehicles m and n . At this point, it is also possible to introduce the average relative speed M , defined as $M_{m,n}$ averaged over all host pairs:

$$M = \frac{1}{N(N-1)/2} \sum_{m,n} M_{m,n} = \frac{1}{N(N-1)/2} \sum_{m=1}^N \sum_{n=m+1}^N M_{m,n} \quad (3.10)$$

where N is the number of vehicles.

- *geographic restrictions*

This metric is very important and, intuitively, can be referred to the notion of *degree of freedom of points in a map*, the number of directions a vehicle can go after reaching that point. Formally defining this metric is not as trivial as it may seem, and authors (Bai et al.) cite this as their future work in (25).

As it concerns the *derived mobility metrics*, they are derived from graph theoretic models as well as other mathematical models and become very important because

¹In their paper, Bai et al. proved that the best value for c_3 is 2, which allows $\bar{RS}$ to differentiate between the different mobility patterns very clearly.

mobility models usually impact the connectivity graph which, in turn, influence the protocol performance. The most important metrics that fall into this category are the *link change rate*, the *link duration* and the *path duration*.

In (39) and (40), authors suggest the *link change rate* as an indicator of topology change: if a link between two hosts is established /severed due to host movement, we consider the state of the link between them up/down. Clearly, the link change rate is the total number of up/down in unit time. The *average link duration* metric(25) is defined as the average link durations over the host pairs that are within each other's transmission range, where the link duration is the interval during which two hosts are within each other's transmission range. The *average path duration*(82), on the other side, can be defined as the interval between the time when the path is set up and the time when the path is broken.

3. MOBILITY MODELS IN VEHICULAR NETWORKS

Chapter 4

Vehicular Network Simulators

In the previous chapter we have underlined the relevance of network simulation in VANETs research, which became a very important and mandatory step in software design before any further implementation. With network simulators it is possible to easily design new protocols and analyze their impact on a large number of network nodes under several traffic and environmental conditions that may reflect in specific mobility and propagation models (cfr. previous chapter).

The most important feature which characterizes simulation is, obviously, the repeatability, which allows to study the overall performances of a given protocol by varying simply the simulation parameters and, obtaining the results in a shorter amount of time with respect to the real testbeds.

Both in academic and industry research, researchers almost always must resort to simulation as the expense of actual deployment would be too high. Unfortunately, there is no standard vehicular networks simulator and the common practice is to generate a mobility trace using a vehicular mobility simulator and then input this trace to the network simulator. Further, the right choice of the mobility simulator is important as performance in vehicular networks highly depends on the connectivity which, in turn, is affected from the nodes' movements.

In this chapter we presents an overview of the most popular vehicular network simulators and mobility simulators, according to their integration component.

4. VEHICULAR NETWORK SIMULATORS

4.1 Introduction to VANET simulation

In order to simulate a VANET we need both the networking and the mobility component. As we already said, in most cases, these two functionalities are provided by two independent simulators so that researchers build a topology and produce a trace of vehicle movements using a mobility simulator. This trace file, which contains all the coordinates of the vehicles at each step of the simulation (which can range from tenths of milliseconds up to seconds) is then fed to the network simulator.

Recently, several integrated simulators - that contain both networking and vehicular mobility components - have been developed, thus allowing feedback between the two components (e.g., TraNS(76) and ASH(7)). However, the common practice in this research field seems to be still the use of two separate simulators.

VANET simulators can be roughly classified as either loosely integrated or tightly integrated. *Loosely-integrated* simulators use separate mobility simulators and network simulators. The mobility simulator generates the mobility of vehicles and records the vehicular movements into trace files. The network simulator imports these trace files, but there is no direct interaction between the two simulators. *Tightly integrated simulators* do not use trace files but rather embed the mobility simulator and network simulator into a single vehicular network simulator. In some cases, the mobility model and network model can communicate, providing feedback from the network simulator to adjust parameters of vehicular movement. For example, in traffic congestion notification systems, the receipt of certain network messages may cause a vehicle to change its path (i.e., take an early exit). In collision avoidance systems, network messages may cause a vehicle to slow down to avoid an accident. This type of feedback is not supported by loose integration of mobility and network simulators. As loosely integrated simulation is currently more prevalent, separate mobility and network simulators will be discussed first, followed by a description of several tightly integrated simulators.

4.2 Mobility simulators

VANET mobility simulators are used to generate traces of the vehicles' motion that can be usually saved and subsequently imported into a network simulator in order to study the performances of the protocol/application. As mentioned earlier, it is important to

generate realistic movement traces in order to rigorously evaluate VANET protocols because the overall performances depend on the connectivity which, in turn, relies on the movement traces.

4.2.1 TSIS-CORSIM

TSIS-CORSIM (Traffic Software Integrated SystemCorridor Simulation⁽⁵⁾) is a powerful commercial traffic simulation package, developed at the University of Florida and funded by the Federal Highway Administration (FHWA). CORSIM is a microscopic simulation model that is especially designed to simulate highways and surface streets and, thus, includes traffic signals and stop signs.

This simulator is made up of two main components, NETSIM for simulating surface streets, and FRESIM for simulating freeways. Unfortunately, TSIS-CORSIM simulator requires Microsoft Windows and Internet Explorer to run. CORSIM can simulate very

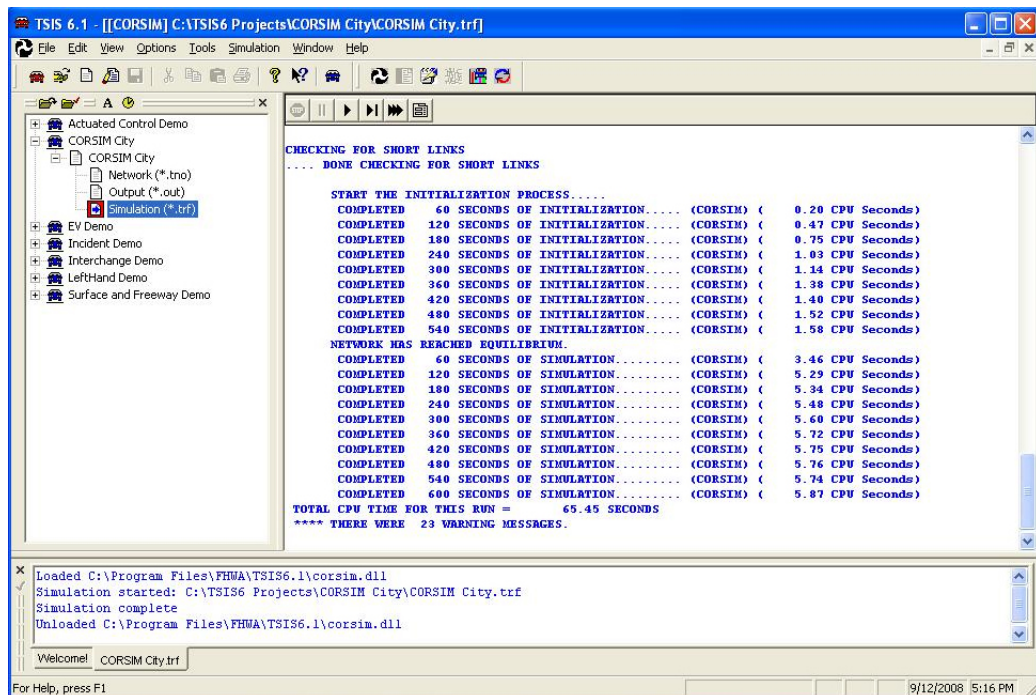


Figure 4.1: CORSIM. A screenshot of CORSIM simulator..

complex traffic scenarios and thus it can take a large amount of information as input. A

4. VEHICULAR NETWORK SIMULATORS

graphical editor, called TRAFED, is included with the package to allow the traffic scenario to be described. The description of highways, or freeways, includes intersections, road segment lengths, number of lanes, merge lane lengths, adding or removing lanes, free flow speeds, roadway curvature, and roadway grade. The description of surface streets, or arterial roadways, includes road segment lengths, lane utilization, free flow speeds, and signal timings. CORSIM also allows the traffic volume to be specified.

The output of this simulator is a MOE (Measure Of Effectiveness) file which is processed into tables that summarize the information. For freeway models, the key metrics produced are throughput, speed, and density. The information for arterial models includes throughput, control delay, and maximum queues. The tables also highlight problem areas that affect arterial and freeway performance, such as ramp intersections. The full TSIS-CORSIM package also includes TRAFVU, which uses CORSIM input and output files to produce animations of the traffic network that has been simulated.

4.2.2 VisSim

VisSim¹ is a commercial simulator developed by PTV Planung Transport Verkehr AG in Karlsruhe, Germany, which allows users to simulate large and complex traffic scenarios, including many different objects like cars, trucks, pedestrians, cyclists, etc.

In detail, VisSim is a *microscopic multi-modal traffic flow simulation software*, which means that each entity of reality that is to be simulated is simulated individually, i.e. it is represented by a corresponding entity in the simulation, thereby considering all relevant properties. The same holds for the interactions between the entities. The opposite would be a macroscopic simulation, in which the description of reality is shifted from individuals to averaged variables like flow and density.

It is also possible to perform to generate 3D animations by using a sophisticated graphical user interface (GUI).

VisSim is only available for Microsoft Windows platforms.

4.2.3 PARAMICS

PARAMICS is a software suite, developed by Quadstone Ltd, which adopts the microsimulation paradigm and runs only on Microsoft Windows based platforms. With

¹<http://www.vissim.com>

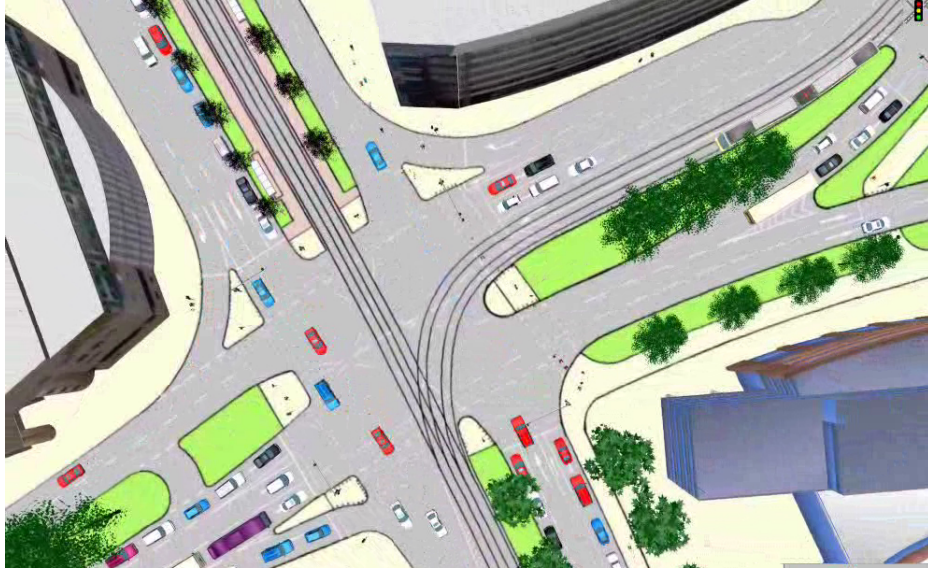


Figure 4.2: VisSim example (A). Simulation of intersections with regard to design alternatives (roundabouts, unsignalized and signal-controlled, grade separated interchanges) and design, test and analysis of vehicle-actuated signal controls.

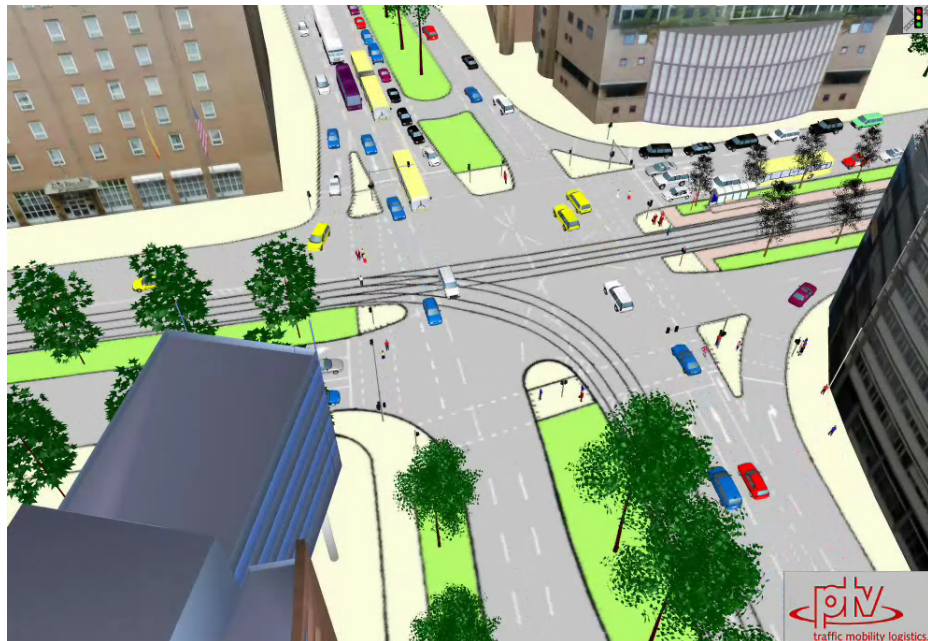


Figure 4.3: VisSim example (B). The same intersection under another perspective.

4. VEHICULAR NETWORK SIMULATORS

PARAMICS it is possible to simulate intersections, highways, urban zones and, like VisSim, it includes a 3D visualization tool.

As already said, PARAMICS is a software suite, which means that it is made up of several modules: modeller, analyser, processor, estimator, designer, converter, programmer and monitor.

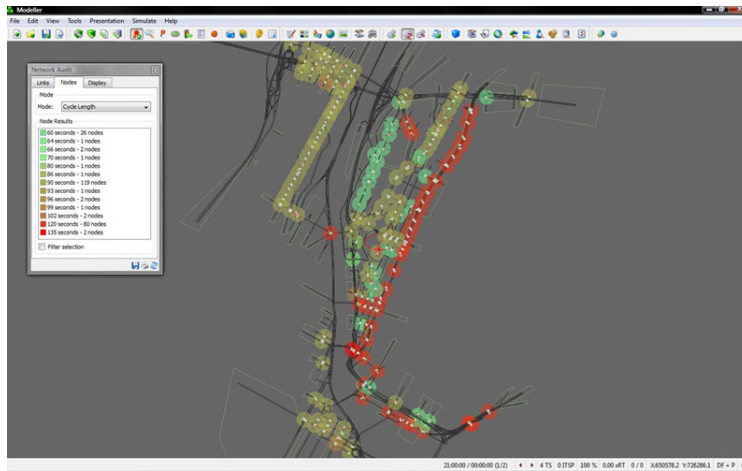


Figure 4.4: Paramics example (A).

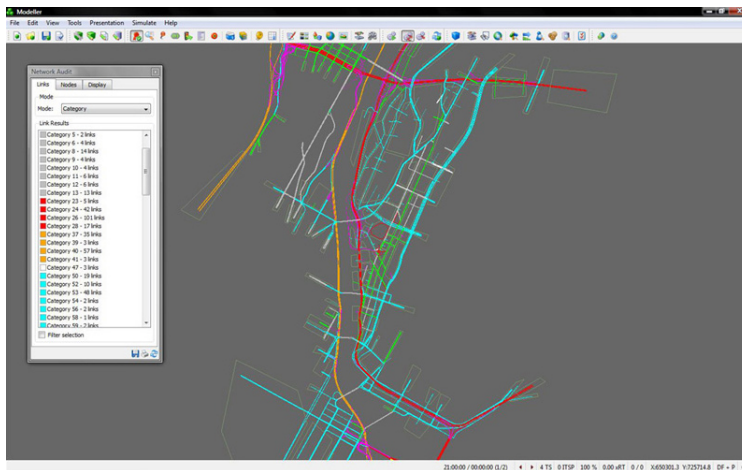


Figure 4.5: Paramics example (B).

4.2.4 Canumobisim and Vanetmobisim

CanuMobiSim(1) is a Java-based simulator designed for mobility simulation by the CANU research group at the University of Stuttgart. It implements several mobility models and can generate mobility traces suitable for use in various network simulators, including ns-2 and GloMoSim. CanuMobiSim allows maps formatted according to the GDF (Geographical Data Files) standard¹ to be imported. Since CanuMobiSim was originally designed for MANET, the mobility models are more suitable for MANETs than VANETs. The mobility models implemented in the CanuMobiSim simulator include the following: *brownian motion*(60), *GaussMarkov random motion*, *incrementally changed random motion*, *random waypoint*(44), *graph-based mobility model*, *constant speed motion*, *fluid traffic model*, *intelligent driver model* (98) and *smooth motion*.

Vanetmobisim(35) is an opensource user mobility simulator developed as an extension to Canumobisim(1). It is completely platform independent as it is coded in Java and it is capable to produce mobility traces for different simulator such as ns-2(3), Glomosim(2) and Qualnet(4).

The aim of Vanetmobisim is to provide a high degree of realism in mobility simulations, thus taking into account both macro and micro mobility features. When considering macro-mobility, it is considered not only the road topology but also the road structure (unidirectional, bidirectional, single/multi-lane), the road characteristic (speed limits, vehicle class based restrictions) and the presence of traffic signs (stop signs, traffic lights, etc.). Moreover, in the macro-mobility it is also considered the effects of the presence of point of interests, which can influence movements patterns. As the road topology is a key factor to obtain realistic results when simulating vehicular movements, Vanetmobisim allows to define the road topology in the following ways (the first two were imported from Canumobisim):

- *GDF map*: the road topology is imported from Geographical Data File. Unfortunately, most of GDF libraries are not freely accessible.
- *TIGER map*: in this case, topology is extracted from a map obtained from the TIGER database(6). The map's details are not as high as GDF but it is open

¹International Standard Organization (ISO), Intelligent transport systems - Geographic Data Files (GDF), overall specification, ISO 14825:2004, 2004.

4. VEHICULAR NETWORK SIMULATORS

and contains digital descriptions of wide urban and rural areas of all districts of United States.

- *Clustered Voronoi graph*: the road topology is randomly generated by creating a Voronoi tessellation¹ on a set of non-uniformly distributed points. This creates fast and configurable random graphs, yet reflecting the non-uniform distribution of obstacles in a urban area.
- *User-defined maps*: the road topology is specified by listing the vertices of the graph and their interconnecting edges.
- *OpenStreetMap*: this feature is included in the unofficial version 2.0.9 of Vanet-mobisim and it allows to import real maps from Openstreetmap website, a free editable map of the whole world. It is also possible to import a selected map area

¹In mathematics, a Voronoi diagram is a special kind of decomposition of a metric space determined by distances to a specified discrete set of objects in the space, e.g., by a discrete set of points. It is named after Georgy Voronoi, also called a Voronoi tessellation, a Voronoi decomposition, or a Dirichlet tessellation (after Lejeune Dirichlet). In the simplest case, we are given a set of points S in the plane, which are the Voronoi sites. Each site s has a Voronoi cell, also called a Dirichlet cell, $V(s)$ consisting of all points closer to s than to any other site. The segments of the Voronoi diagram are all the points in the plane that are equidistant to the two nearest sites. The Voronoi nodes are the points equidistant to three (or more) sites.

A point location data structure can be built on top of the Voronoi diagram in order to answer nearest neighbor queries, where one wants to find the object that is closest to a given query point. Nearest neighbor queries have numerous applications. For example, when one wants to find the nearest hospital, or the most similar object in a database. A large application is vector quantization, commonly used in data compression. With a given Voronoi diagram, one can also find the largest empty circle amongst a set of points, and in an enclosing polygon; e.g. to build a new supermarket as far as possible from all the existing ones, lying in a certain city. The Voronoi diagram is useful in polymer physics. It can be used to represent free volume of the polymer. It is also used in derivations of the capacity of a wireless network. In climatology, Voronoi diagrams are used to calculate the rainfall of an area, based on a series of point measurements. In this usage, they are generally referred to as Thiessen polygons. Voronoi diagrams are used to study the growth patterns of forests and forest canopies, and may also be helpful in developing predictive models for forest fires. Voronoi diagrams are also used in computer graphics to procedurally generate some kinds of organic looking textures. In autonomous robot navigation, Voronoi diagrams are used to find clear routes. If the points are obstacles, then the edges of the graph will be the routes furthest from obstacles (and theoretically any collisions). In computational chemistry, Voronoi cells defined by the positions of the nuclei in a molecule are used to compute atomic charges. This is done using the Voronoi deformation density method (source [http : www.wikipedia.org](http://www.wikipedia.org)).

directly into the simulator GUI. This is the option we used to simulate our data dissemination protocol, CORP.

Anyways, the road topology is implemented as a graph over whose edges the vehicle movements are constrained.

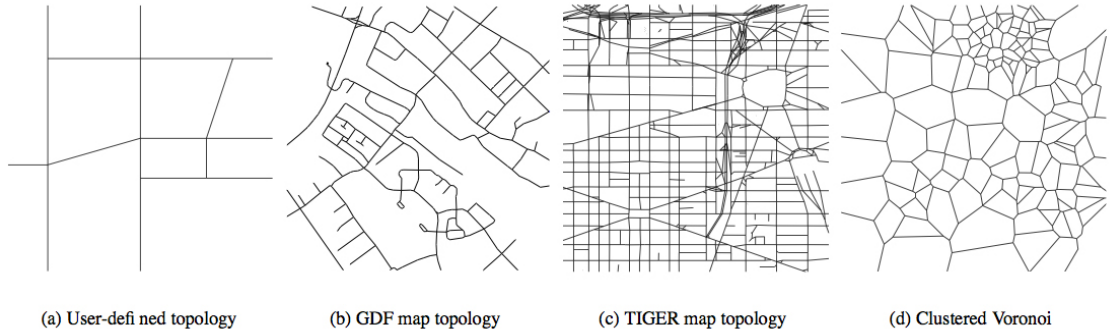


Figure 4.6: Vanetmobisim road topology example.

It is important to recall that Vanetmobisim introduces new feature on road characterization like:

- introduction of roads with multiple lanes in each direction;
- physical separation of opposite traffic flows on each road;
- definition of independent speed limits on each road of the topology;
- implementation of traffic signs at each road intersection. By default, traffic intersections are regulated by stop signs, forcing vehicles to stop and wait for free road before crossing. In addition (but it is still experimental), it is possible to use traffic lights, whose temporization is customizable.

The vehicle movements in Vanetmobisim are very realistic as it exploits the capability of building up movements patterns from the cooperation of a trip generation module, which defines the sets of points of interests, and a path computation module, whose task is to compute the best path between those points.

The trip generation module is mainly based on two options: the first is a *random trip*, where the start and stop points of movement patterns are randomly selected

4. VEHICULAR NETWORK SIMULATORS



Figure 4.7: Vanetmobisim road topology example imported from OpenStreetMap database. The map showed in this picture ("Quarto Miglio" district in Rome) was exported in xml-format from OpenStreetMap and then imported into Vanetmobisim simulator. The red dots represent the constrained vehicles moving to across roadmap during the simulation.

among the vertices of the graph representing the road topology; the second is an *activity sequences generation*, as a set of start and stop points are explicitly provided in the road topology description, and cars are forced to move among them. Independently from the trip generation method employed, the path computation, i.e. the selection of the best sequence of edges to reach the selected destination, can be performed in three ways:

1. *Dijkstra's algorithm*: the shortest path to destination is evaluated with the Dijkstra's algorithm, with edges' cost inversely proportional to their length;
2. *traffic congestion level*: method does not only considers the length of the path, but also the traffic congestion level, by weighting the cost of traversing an edge also on the number of cars traveling on it, thus modeling the real world tendency of drivers to avoid crowded paths;
3. *road speed limit*: The last method, which is not present in the original CanuMobisim, extends the other two, by also accounting for the road speed limit when calculating the cost of an edge, in a way that fastest routes are preferred.

Of course, when the definition of vehicular movement paths is a factor of interest in the mobility simulation, it is possible to use a combination of trip generation and path computation methods offers a wide range of possibilities.

4.2.4.1 Microbility features in Vanetmobisim

At this point, it is important to come again on the micro-mobility features included in vanetmobisim, as they play the main role in the generation of realistic vehicular movements and they can, thus, making the difference in the simulation results. It is possible to distinguish three classes of micro-mobility models, featuring an increasing degree of detail, mainly based on whether the individual speed of vehicles is computed: in a *deterministic way*, as a *function of nearby vehicles in a single lane scenario*, as a *function of nearby vehicles' behavior in a multi-flow interaction scenario* (i.e. like the urban one). The first three models we can find in Vanetmobisim (inherited from CanuMobiSim) are the *Graph-Based Mobility Model (GBMM)(96)*, the *Constant Speed Motion (CSM)(1)* and the *Smooth Motion Model (SMM)(14)*. All these models fall into the first class, as the speed of each vehicle is determined on the basis of the local state of each car while any external effect is ignored. Both models constrain a random movement of nodes on a graph, possibly including pauses at intersections (CSM) or smooth speed changes when reaching or leaving a destination (SSM). The movements is random, as the vehicles select a destination and move towards it along a shortest length path (ignoring other vehicles and overlapping). Obviously, these models fail to reproduce realistic movements of vehicles' clusters.

In the second class, instead, there are two main models: the *Fluid Traffic Model (FTM)(88)* and the *Intelligent Driver Model(98)*. Both models fall this time into the second class, as they take into account the presence of nearby vehicles when calculating the relative speeds (but do not consider, however, the case in which multiple vehicular flows have to interact in presence of intersections). In particular, the FTM describes the speed as a monotonically decreasing function of the vehicular density, forcing a lower bound on speed when traffic congestion reach a critical state according to the following equation:

4. VEHICULAR NETWORK SIMULATORS

$$s = \max \left[s_{\min}, s_{\max} \left(1 - \frac{k}{k_{jam}} \right) \right] \quad (4.1)$$

where s is the output speed, $s_{\min}$ and $s_{\max}$ are respectively the minimum and the maximum speed, k_{jam} is the vehicular density for which a traffic jam is detected, and k is the current vehicular density of the road the node is moving on. The parameter k is calculated as n/l , where n is the number of cars on the road and l is the road segment itself.

The IDM model, instead, relies mainly on the following car (that is why this model falls into the so called "*car following model*") and the instantaneous acceleration of a vehicle is computed according to the formula:

$$\frac{dv}{dt} = a \left[1 - \left(\frac{v}{v_0} \right)^4 - \left(\frac{s^*}{s} \right)^2 \right] \quad (4.2)$$

$$s^* = s_0 + \left(vT + \frac{v\Delta v}{2\sqrt{ab}} \right) \quad (4.3)$$

In the equation (4.2), v is the vehicle's current speed, v_0 the desired velocity, s is the distance from the preceding car and s_0 is the so called *desired dynamical distance*, computed in (4.3) as a function of the minimum bumper-to-bumper distance s_0 , the minimum safe time headway T , the speed difference with respect to front vehicle's velocity Δv , and the maximum acceleration values a and b .

Vanetmobisim, as said before, inherit alle these models from CanuMobiSim but, in addition, we can find two IDM variants: the *Intelligent Driver Model with Intersection Management (IDM-IM)* and the *Intelligent Driver Model with Lane Changes (IDM-LC)*. According to the first, vehicles moving in the scenario can handle, basically, two kinds of intersections: crossroads regulated by stop signs and road junction ruled by traffic lights. Clearly, even if the intersection modeling is applied only to the first car on each road, the overall effect of handling such a scenario is reflected to the other vehicles as IDM automatically adapts the behavior of car following the leading one. Everytime the vehicle doesn't find any other vehicle between itself and the intersection regulated by stop signs, the following parameters are used:

$$\begin{cases} s = \sigma - S \\ \Delta v = v \end{cases} \quad (4.4)$$

where σ represents the current distance to the intersection and S is a safety margin, which considers the gap between the center of the intersection and the point where the car would actually stop. Once a car is halted at a stop sign, it is informed by the macroscopic level description of the number of cars already waiting to cross the intersection from any of the incoming roads. If there are no other cars, the vehicle may pass. Otherwise, it has to wait for its turn in a first-arrived-first-passed and right hand rule policy. When a vehicle is heading towards a traffic light intersection, it is informed by the macroscopic description about the state of the semaphore. If the color is green, passage is granted and the car maintains its current speed through the intersection. If the color is red, crossing is denied and the car is forced to decelerate and stop at the road junction by using the modied IDM parameters similarly to a stop sign. It may also be stressed out that vehicles behavior can dynamically vary in presence of traffic lights, according to red-to-green and green-to-red switches. In the former case, a car currently decelerating to stop at a red light will accelerate again if the semaphore turns green before it has completely halted. In the latter case, a vehicle keeping its pace towards a green light will try to stop if the light becomes red before it has passed through the intersection. A minimum breaking distance $\bar{s}$ is evaluated by means of simple kinematic formulae as the following:

$$\bar{s} = vt - \frac{kb}{2}t^2 = v \left(\frac{v}{kb} \right) - \frac{kb}{2} \left(\frac{v}{kb} \right)^2 = \frac{v^2}{2kb} \quad (4.5)$$

which describes the space needed to come to a full stop as a function of the current speed v of the vehicle, v , the time t and the deceleration value, kb . The last parameter is the maximum safe deceleration, evaluated as the IDM comfortable braking value b scaled by a factor $k \geq 1$.

Upon computation of $\bar{s}$, if the vehicle finds that it is not possible to stop before the intersection, even braking as hard as possible (i.e., if $\bar{s} > \sigma - S$), then it crosses the intersection at its current speed. Otherwise, it stops by applying a strong enough deceleration. This reproduces a real world simulation, since when a traffic light switches to red, drivers only stop if safety braking conditions can be respected.

The second variant, the IDM-LC, extends the original IDM model introduced in Canu-MobiSim with the possibility for vehicles to change lane and overtake each others, taking advantage of the multi-lane capability. The multi-lane handling is a trivial task especially when approaching an intersection; in that case, a vehicle can decide to keep

4. VEHICULAR NETWORK SIMULATORS

traveling on the same lane in the next road (if the same lane is present on the new segment) or try to merge to its right (if the current lane is not present in the new road). In the latter situation, if the lane on the right is congested, the vehicle has to stop and wait until a spot becomes available. As it concerns the overtaking mechanism, the IDM-LC inherits the MOBIL model(97), based on a game theoretical approach to address the lane changing problem; basically, it allows a vehicle to change lane if this shift minimize the overall braking. Mathematically speaking, this is achieved when the following two conditions are satisfied:

$$a^l - a \pm a_{bias} > p \left(a_{cur} + a_{new} - a_{cur}^l - a_{new}^l \right) + a_{thr} \quad (4.6)$$

$$a_{new}^l > -a_{safe} \quad (4.7)$$

This means that a vehicle is allowed to move to lane l if the disadvantage of the driver who is doing the change is greater than the disadvantages of the following cars (to be more precise, when the first inequality is satisfied) in the current ($a_{cur} - a_{cur}^l$) and in the candidate ($a_{new} - a_{new}^l$) lanes. The p factor is the *driver's politness*, ranging from above 1 (very polite driver) to 0 (selfish or even malicious driver); the a_{thr} is the threshold acceleration and represents a the minimum acceleration advantage to allow lane change (this is in order to avoid lane hopping in border cases). The last two parameters, a_{bias} and a_{safe} , represent respectively the real tendency to stay on the right most lane and the condition that the driver can change lane only if the back vehicle does not to brake too hard (its deceleration has to be over the value a_{safe}).

4.2.5 SUMO and MOVE

SUMO is a microscopic, space continuous and time discrete traffic simulator written in C++ capable to provide accurate and realistic mobility patterns. The project started as an open source project in 2001 with the goal to support the traffic research community with a common platform to test and compare models of vehicle behaviour, traffic light optimisation, routing etc.

In SUMO, each vehicle has an own route and is simulated individually according to a fast but still accurate model developed in 1998 by Stefan Krauss (51) (extended by some further assumptions), which uses discrete time steps of 1s. SUMO is very fast, around 100.000-200.000 vehicles can be simulated in real time on a desktop PC, including:

- *collision free vehicle movement;*
- *several vehicle types;*
- *simulation of traffic lights;*
- *junction-based right-of-way rules* (junctions with streets having equal / different priorities, e.g. right-before-left);
- *lane changing;*
- *lane-to-lane connections;*

MOVE (46) is an extension to SUMO that adds a GUI for describing maps and defining vehicle movement and allows the user to import Google Earth maps. MOVE also includes a visualization tool that allows users to view the generated mobility trace. Basically, MOVE is composed of two components: the road map editor and the vehicle movement editor. The former serves to manually and randomly generate a road map, either from TIGER/line files or Google earth files, whereas the latter allows to specify the properties of each vehicle, like the maximum speed, the acceleration, the probability of turning at crossroads, the path to take etc. The information collected by the two editors are sent to the SUMO compiler, then a trace file in ns-2 or Qualnet format is generated. MOVE has been compared by simulation to RWP using AODV. The results show that MOVE causes low reception rate.

4.3 Network simulators

In this section we describe the most used network simulators in the research field. These applications allow to simulate (at different level of detail) the ISO-OSI layers, taking into account the propagation and fading effect of the radio signals.

As described earlier, a mobility simulator is generally used to produce node movement traces that are then fed to the network simulator. The network simulator then controls the communications between the mobile nodes. As these network simulators support wireless communication, most of them include at least a simple node mobility model,

4. VEHICULAR NETWORK SIMULATORS

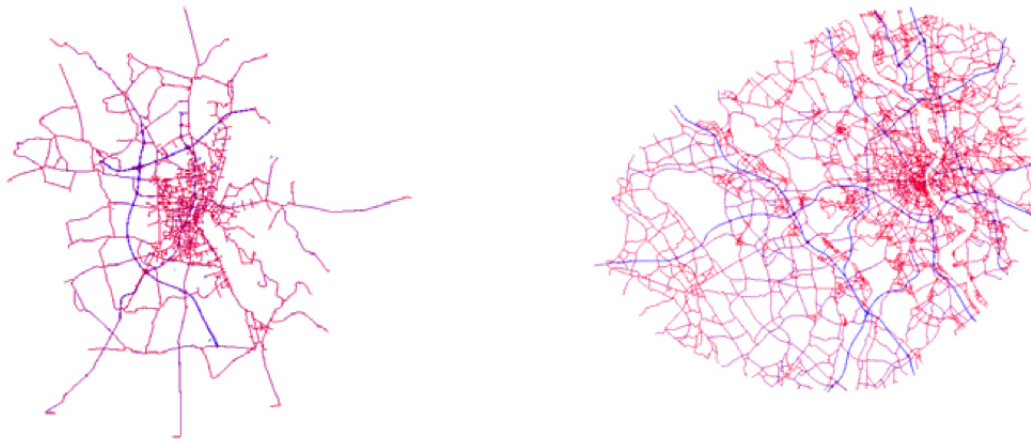


Figure 4.8: SUMO simulated network example. Two example networks as simulated by SUMO; left: the city of Magdeburg from the INVENT-project; right: the area around the city of Cologne.

which includes the following models: *Random Drunken Model*¹, *Random Waypoint Model*, *Trace file*².

4.3.1 OPNET

OPNET³ is a commercial network simulator used for simulations of both wired and wireless networks. It supports a wide range of wireless technologies such as MANETs, IEEE 802.11 wireless LANs, WiMAX, Bluetooth, and satellite networks. OPNET provides a graphical editor interface to build models for various network entities from physical layer modulator to application processes, and includes graphical packages and libraries for presenting simulation scenarios and results. There are three basic phases of the OPNET deployment process. First, choose and configure node models to use in simulations, such as a wireless node, trajectory, and so on. Second, build and organize the network by setting up connections for different entities. Third, select the desired statistics (local or global) to collect during the simulation.

In order to setup a simulation, there are three main configuration files: *network config-*

¹At each intersection, the vehicle randomly selects from four directions or chooses to remain stationary.

²Vehicles' mobility is based on a trace file that can be imported into the simulator.

³Website: <http://www.opnet.com>

uration file (nodes, routers, applications, protocols, radio settings), *node configuration file* (data related to mobile nodes such as IP addresses) and *global parameter file* (simulation time, coordinate system, random seed, protocol stack, statistic filter).

4.3.2 GloMoSim and QualNET

GloMoSim (112) is a network simulator developed at UCLA in 1999 especially designed of MANET and includes a large set of routing protocols and several physical layer implementations. Being an integrated network simulator, it offers also the possibility to generate trace according to some mobility models like, i.e. Random Waypoint, Random Drunken (and, of course, real traces import).

The software was dismissed in 2000 but it is still possible to download only for educational purposes. On the other side, the commercial version of GloMoSim became Qualnet simulator, developed by Scalable Network Technologies.

Qualnet¹ is an extremely powerful and detailed network simulator which provides a large set of wireless physical and MAC layers models and mobility models, allowing reseearchers to develop and and easily simulate protocols for sensor networks, satellite and military warfare. Another important aspect of Qualnet, is the avaiability for Windows and Unix/Linux platforms.

QualNet delivers high fidelity simulations of network devices, transmitters, antennas, terrestrial characteristics, and human interactions, all at real time speed or greater thanks to an efficient parallel kernel and patent-pending, computationally-efficient code, making possible to get the same accurate simulations of wireless and wired networks for 50, 500, or 5,000 nodes. Models in source form provide developers with a solid library on which to build and experiment with new network functionality. The end result is accurate prediction of network performance for a diverse set of application requirements and uses. From wired LANs and WANs, to cellular, satellite, WLANs and mobile ad hoc networks, QualNet's library is extensive and because of its efficient kernel, QualNet models large scale networks with heavy traffic and mobility in reasonable simulation times.

Qualnet Simulator features can be summarized as the follows:

- *instant playback of simulation results to minimize unnecessary model executions;*

¹<http://www.scalable-networks.com>

4. VEHICULAR NETWORK SIMULATORS

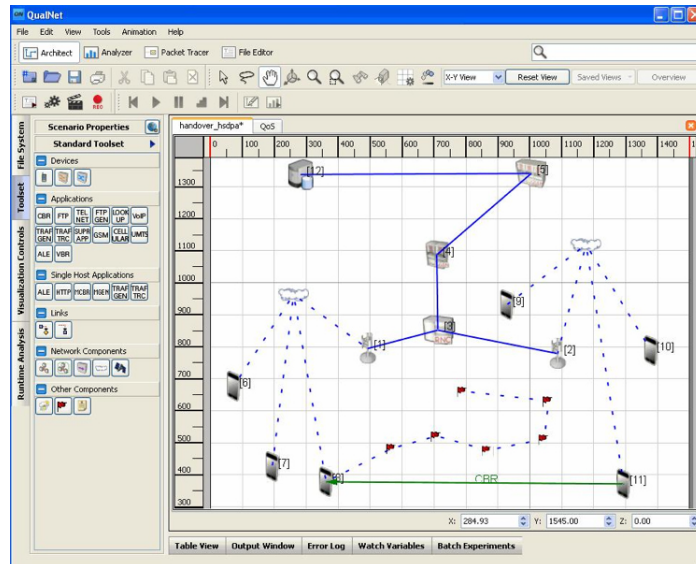


Figure 4.9: Qualnet simulator. Qualnet in visual mode.

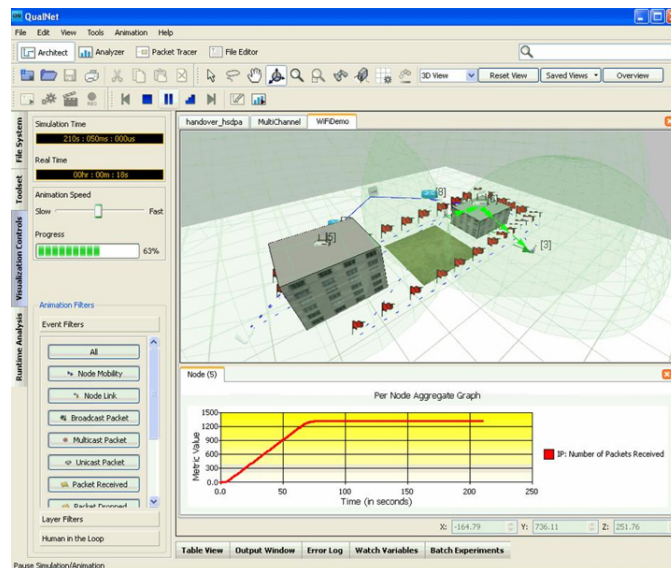


Figure 4.10: Qualnet simulator. Qualnet in visual mode, additional example.

- *fast simulation results for thorough exploration of model parameters;*
- *fast model set up with a powerful Graphical User Interface (GUI) for custom code development and reporting options;*
- *scalable up to tens of thousands of nodes;*
- *real-time simulation for man-in-the-loop and hardware-in-the-loop models;*
- *multi-platform support;*

and it consists of several modules which handle specific functions like:

- *Scenario Designer*

It is a graphical tool that provides an intuitive model set up capability and is used to create and design experiments in QualNet. The Scenario Designer enables a user to define the geographical distribution, physical connections and the functional parameters of the network nodes, all using intuitive click and drag tools, and to define network layer protocols and traffic characteristics for each node.

- *Animator*

This module is used to execute and animate experiments created in the Scenario Designer. Using the Animator a user can watch traffic flow through the network and create dynamic graphs of critical performance metrics as a simulation is running.

- *Packet Tracer*

It is a packet-level visualization tool for viewing the contents of packets as they travel up and down the protocol stack.

- *Analyzer*

Statistical graphing tool that displays network statistics generated from a QualNet experiment. Using the Analyzer, a user can view statistics as they are being generated, as well as compare results from different experiments.

Due to its powerful features and scalability, we decided to use Qualnet in our research on data dissemination with rateless codes in vehicular networks.

4. VEHICULAR NETWORK SIMULATORS

4.3.2.1 Discrete event simulation

QualNet is a discrete-event simulator. In discrete-event simulation, a system is modeled as it evolves over time by a representation in which the system state changes instantaneously when an event occurs, where an event is defined as an instantaneous occurrence that causes the system to change its state or to perform a specific action. Examples of events are: arrival of a packet, a periodic alarm informing a routing protocol to send out routing update to neighbors, and so on. Examples of actions to take when an event occurs are: sending a packet to an adjacent layer, updating state variables, starting or restarting a timer, etc.

In discrete-event simulation, the simulator maintains an event queue. Associated with each event is its event time, i.e., the time at which the event is set to occur. Events in the event queue are sorted by the event time. The simulator also maintains a simulation clock which is used to simulate time. The simulation clock is advanced in discrete steps, as explained below.

The simulator operates by continually repeating the following series of steps until the end of simulation:

1. the simulator removes the first event from the event queue, i.e., the event scheduled for the earliest time;
2. the simulator sets the simulation clock to the event time of the event. This may result in advancing the simulation clock;
3. the simulator handles the event, i.e., it executes the actions associated with the event. This may result in changing the system state, scheduling other events, or both. If other events are scheduled, they may be scheduled to occur at the current time or in the future.

4.3.2.2 Qualnet protocol stack

One of the best feature in Qualnet (very appreciated from researchers) is the protocol stack's implementation, a layered architecture similar to that of the TCP/IP network protocol stack. According to this architecture, data moves between adjacent layers (from top to bottom): Application, Transport, Network, Link (MAC) and Physical Layers. Adjacent layers in the protocol stack communicate via well-defined APIs, and

generally, layer communication occurs only between adjacent layers. For example, Transport Layer protocols can get and pass data to and from the Application and Network Layer protocols, but cannot do so with the Link (MAC) Layer protocols or the Physical Layer protocols. This rule concerning communication only between adjacent layers is not a limit of the simulator because it can be also circumvented by the programmer.

Each protocol operates at one of the layers of the stack. Protocols in QualNet essentially operate as a finite state machine. The occurrence of an event corresponds to a transition in the finite state machine. The interface between the layers is also event based. Each protocol can either create events that make it change its own state (or perform some event handling), or create events that are processed by another protocol. To pass data to, or request a service from, an adjacent layer, a protocol creates an event for that layer.

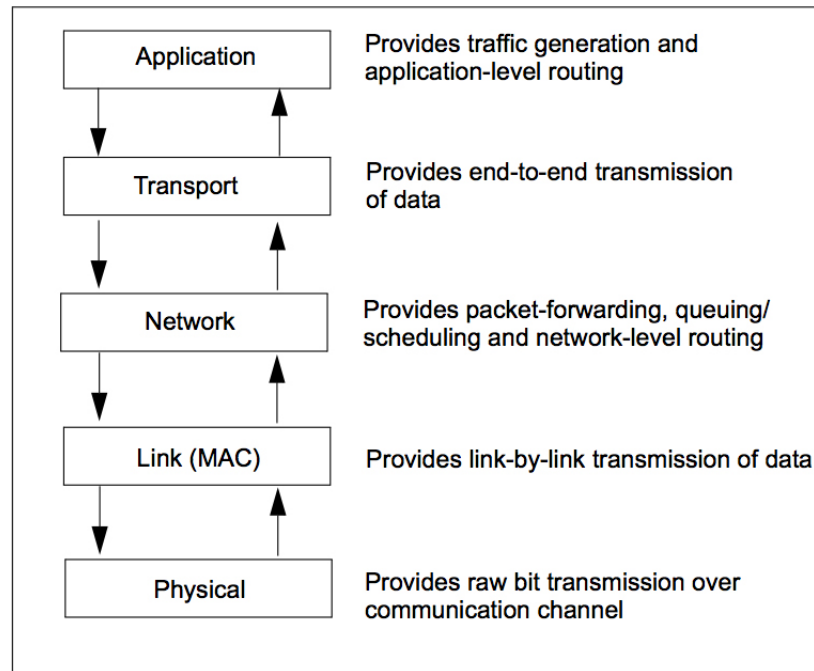


Figure 4.11: Qualnet protocol stack. This picture shows the simulator protocol stack and the general functionality of each layer.

- *Application Layer*

The Application Layer is responsible for traffic generation and application level

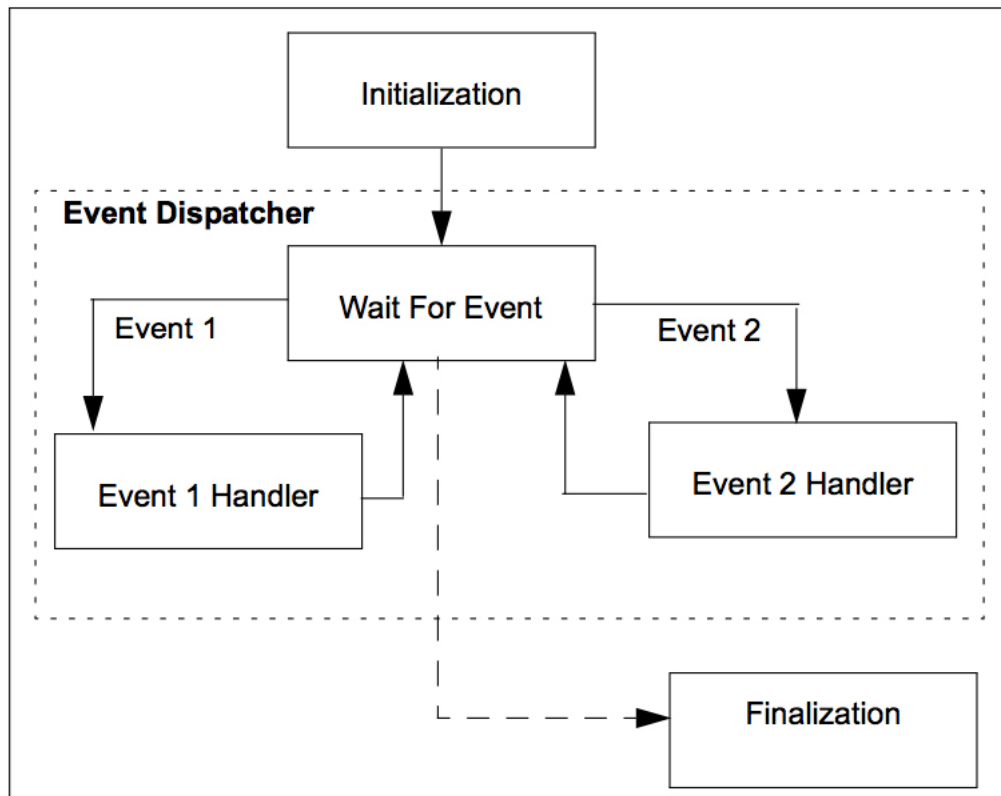


Figure 4.12: Qualnet protocol model. The picture shows the finite state machine representation of a protocol in QualNet. At the heart of a protocol model is an Event Dispatcher, which consists of a Wait For Event state and one or more Event Handler states. In the Wait For Event state, the protocol waits for an event to occur. When an event for the protocol occurs, the protocol transitions to the Event Handler state corresponding to that event (e.g., when Event 1 occurs, the protocol transitions to the Event 1 Handler state). In this Event Handler state, the protocol performs the actions corresponding to the event, and then returns to the Wait For Event state. Actions performed in the Event Handler state may include updating the protocol state, or scheduling other events, or both.

routing. Protocols written at the Application Layer rely on the Transport Layer to deliver application-level data from the source to the destination. Thus, Application Layer protocols pass data down to the Transport Layer at the source node, and receive data from the Transport Layer at the destination node. Examples of traffic-generating Application Layer protocols implemented in QualNet are Constant Bit Rate (CBR), FTP, and Telnet. Examples of Application Layer routing protocols implemented in QualNet are RIP, Bellman-Ford, and BGP.

- *Transport Layer*

The Transport Layer provides end-to-end data transmission services to the Application Layer. Protocols written at the Transport Layer receive data from the Application Layer and rely on the Network Layer for data forwarding at the source node, and receive data from the Network Layer and pass data to the Application Layer at the destination node. Examples of Transport Layer protocols include UDP, TCP and RSVP-TE.

- *Network Layer*

The Network Layer is responsible for data forwarding and queuing/scheduling. The Internet Protocol (IP) resides at this layer and is responsible for packet forwarding. At the source node, the Network Layer receives data from the Transport Layer and relies on the Link (MAC) Layer for link-by-link data delivery. At the destination node, the Network Layer receives data from the Link (MAC) Layer and passes the data up to the Transport Layer. The Network Layer also implements certain types of routing protocols. Examples of Network Layer routing protocols implemented in QualNet are AODV, DSR, OSPF, and DVMRP. Examples of queuing/scheduling protocols implemented in QualNet are FIFO, RED, RIO, WFQ, and WRR.

- *Link (MAC) Layer*

The Link (MAC) Layer provides link-by-link transmission. At the sending side, the Link (MAC) Layer receives data from the Network Layer and passes the data to the Physical Layer for transmission over the wired or wireless channel. At the receiving side, the Link (MAC) Layer receives data from the Physical Layer and forwards the data up to the Network Layer. Examples of protocols at the Link

4. VEHICULAR NETWORK SIMULATORS

(MAC) Layer implemented in QualNet are point-to-point, IEEE 802.3, IEEE 802.11, and CSMA.

- *Physical Layer*¹

The Physical Layer is responsible for transmitting and receiving raw bits from the wired and wireless channel. At the source node, the Physical Layer receives data from the Link (MAC) Layer and sends the data to the Physical Layer of the destination node. At the destination node, the Physical Layer receives data from the Physical Layer of the source node and passes the data to the Link (MAC) Layer. Examples of Physical Layer protocols implemented in QualNet are wired point-to-point links, IEEE 802.3, and IEEE 802.11.

- *The Communication medium*

The communication medium transmits signals between nodes. It interfaces with the Physical Layer entities at the nodes. A wireless communication medium model in QualNet simulates the propagation of signals between nodes, taking into account both propagation delays and signal attenuation due to path loss, fading, and shadowing.

In QualNet, a communication medium model has three components: a path loss model, a fading model, and a shadowing model. Path loss models in QualNet include free space, two ray, and Irregular Terrain Model (ITM). QualNet implements the Ricean fading model. Rayleigh fading is a special case of Ricean fading. QualNet provides models for two shadowing models: constant and lognormal.

4.3.3 NS-2

NS-2 is an open-source discrete event network simulator that supports both wired and wireless networks, including many MANET routing protocols and an implementation of the IEEE 802.11 MAC layer and it is the most widely used simulator for academic networking research. There are implementations of several mobility models available for NS-2, including *Random Trip Mobility*(73) and *Semi-Markov Smooth Mobility*(114). NS-2 simulates the wireless physical layer and the important parameters that influence its behavior (e.g., channel fading).

The core of NS-2 is written in C++, but users interact with NS-2 by writing TCL

¹For wired networks, the Physical Layer code is incorporated into the Link (MAC) Layer.

scripts. which should contain all of the commands needed to run the simulation (e.g., setting up the topology, specifying wireless parameters, and so on). As with QualNet, several of the mobility simulators can generate node descriptions and movement traces suitable for use in NS-2.

A typical wireless NS-2 simulation produces an event trace file and an animation trace file, used by the included utility `nam` to provide animation of the simulation. The event trace file includes packet enqueue (transmission), packet dequeue (forwarding), packet drops, and packet reception.

4.3.4 J-Sim

J-Sim is an open-source simulation environment, developed entirely in Java. J-Sim provides two mobility models: trajectory-based and random waypoint. J-Sim is presented as an alternative to ns-2, because it is designed to be easier to use. In J-Sim, applications are built as a set of components that can be designed and tested separately. J-Sim can take a TCL file as input, similar to ns-2, but with a different format. Like ns-2, J-Sim produces an event trace file and an animation file, suitable for use in `nam`.

4.3.5 OMNeT++

OMNeT++¹ is an open-source simulation environment. The primary simulation applications are Internet simulations, mobility, and ad hoc simulations. OMNeT++ has a component-based design, meaning that new features and protocols can be supported through modules. OMNeT++ supports network and mobility models through the independently developed Mobility Framework and INET Framework modules. Simulation design in OMNeT++ is GUI-based, and output data can be plotted through the GUI as well. OMNEST² is the commercial version of OMNeT++, offered by Simulcraft, Inc.

4.3.6 SWANS

SWANS(13) (Scalable Wireless Ad hoc Network Simulator) was developed to be a scalable alternative to ns-2 for simulating wireless networks. Based on comparisons of SWANS, GloMoSim, and ns-2, (45) SWANS was determined to be the most scalable

¹OMNeT++ website: <http://www.omnetpp.org/>

²OMNEST website: <http://www.omnest.com/>

4. VEHICULAR NETWORK SIMULATORS

and the most efficient in memory usage with the fastest runtime. Along with better performance, SWANS delivered similar results as ns-2, at least for the network components that were implemented in both. The input for SWANS is a Java file that creates the nodes and specifies how these nodes should move (the node movement scenario) and how they should communicate (the communication scenario). The user can select any of the ready-made applications in SWANS and associate it with any node(s) to execute it at the node application layer. Also, SWANS gives the user the flexibility to build a custom application and execute it at the application layer of any node.

4.4 Tightly Integrated Simulators

As mentioned in 4.1, the simulation of VANET applications not only requires simulating the wireless communication between the vehicles, but also requires simulating the mobility of the vehicles. Unfortunately, these two aspects of VANET simulation have often been decoupled. Both vehicular mobility and wireless communication have large communities concerned with their modeling and simulation, so high-quality simulators exist in each of these areas, as discussed in the previous sections. The problem is how to merge the two types of simulators (network simulator and mobility simulator).

A simple method to achieve this goal is to implement mobility models in a network simulator, but without allowing the network messages to feed back to the mobility model. This kind of simulation is called one-way communication (from mobility model to network). These types of simulators are suitable for simulating infotainment-related VANET applications, including Internet connectivity, multimedia applications, and peer-to-peer applications, where the communication does not affect vehicles movements. In contrast, tightly integrated simulators, that offer two-way communication, usually consist of two sub-simulators (network and mobility) which can communicate with each other. These simulators are more appropriate for safety-related and traffic information applications that assume that feedback from the network will affect vehicles movements. In these types of applications, the traffic simulator feeds the network simulator with position information, speed, acceleration, direction, and so on. The VANET application that runs at the top level of the network simulator incorporates this information with surrounding vehicles information in order to notify the driver of upcoming congestion or a possible collision. Based on this notification, driving decisions (i.e., vehicle

mobility) may be affected. For example, in a congestion notification system, the driver may choose to change lanes or take a different path. These decisions need to propagate back to the mobility simulator to be reflected in the vehicle mobility information.

Usually, any simulator has an events queue to store the events that should be executed according to their scheduled execution time. In case of two-way communication simulators, each sub-simulator has its own events queue; these two events queues can be combined into one events queue, or they can be kept separate, which implies that extra overhead will be needed for synchronization. Based on this decision, the two-way communication simulators are separated into two categories: *those with a single events queue* and *those with two events queues*. Having a single events queue can be achieved through implementing one of the sub-simulators in the other. Often, the vehicular mobility sub-simulator is implemented in the network sub-simulator, as in ASH (7) or the two simulators can be highly integrated together, as in NCTUns (99). The advantages of having a single events queue are that the vehicles mobility events and the network events will be inserted in the same queue, which removes the burden of synchronizing the two types of events. In addition, the simulation will be more efficient from the execution time and memory consumption perspectives. The main disadvantage of having a single events queue is that the process of maintaining and extending such simulators is not easy. With two events queues, two-way communication is achieved through an interface that is implemented between the network sub-simulator and the mobility sub-simulator itself. The main function of that interface is to update each sub-simulator with the recent events in the other sub-simulator and, moreover, it synchronizes the event execution in each of the events queues. The main disadvantages are that these types of simulators consume more memory and execution time.

In this section, tightly integrated simulators with one-way communication will be discussed first. However, only those simulators that attempt to package the mobility and network simulator in a single program are included, rather than those that manually feed the mobility simulators movement trace to the network simulator. Next will come an overview of tightly integrated simulators with two-way communication and two events queues, and finally tightly integrated simulators with two-way communication and a single events queue.

4. VEHICULAR NETWORK SIMULATORS

4.4.1 SWANS++

SWANS++¹ extends the network simulator SWANS by adding a GUI to visualize the scenario and a mobility model, STRAW (21) for the vehicles movement in street scenarios. STRAW uses the simple random waypoint mobility model, but it restricts the vehicles movement to real street boundaries, loaded from TIGER/Line data files. STRAW consists mainly of three components: intrasegment mobility, intersegment mobility, and route management and execution. In intersegment mobility, the vehicles move according to a car-following model and change their speed only in certain situations:

1. when the vehicle arrives at an intersection and the next segment is full, the vehicle stops until the next segment has a free slot;
2. when the vehicle has a vehicle in front of it, the vehicle adjusts its speed accordingly in order to maintain a certain distance in between;
3. when the vehicle arrives at a traffic control or stop sign;
4. when the vehicle makes a turn.

For intersegment mobility, according to the system design, there is either a traffic control sign or a stop sign at each intersection that forces the vehicle to alter its speed. The mobility model implemented in STRAW (and therefore, SWANS++) does not support lane changing. The route management and execution (RME) module is responsible for determining the vehicles routes during the simulation. The RME has two techniques to fulfill its task. The first technique is simple intersegment mobility (simple STRAW) at which the vehicles next segment is determined stochastically, while the second technique is origindestination mobility (STRAW OD), in which the vehicles route is predetermined based on the shortest path between the origin and destination. SWANS++ is a tightly integrated simulator, but it does not provide feedback between the mobility and networking modules.

¹SWANS++ is available at <http://www.aqualab.cs.northwestern.edu/projects/swans++/>

4.4.2 GrooveNet

GrooveNet¹ (63) (originally known as GrooveSim (64)) is an integrated network and mobility simulator that allows communication between real and simulated vehicles. Originally, GrooveNet extended the open-source simulator *RoadNav*² by adding a

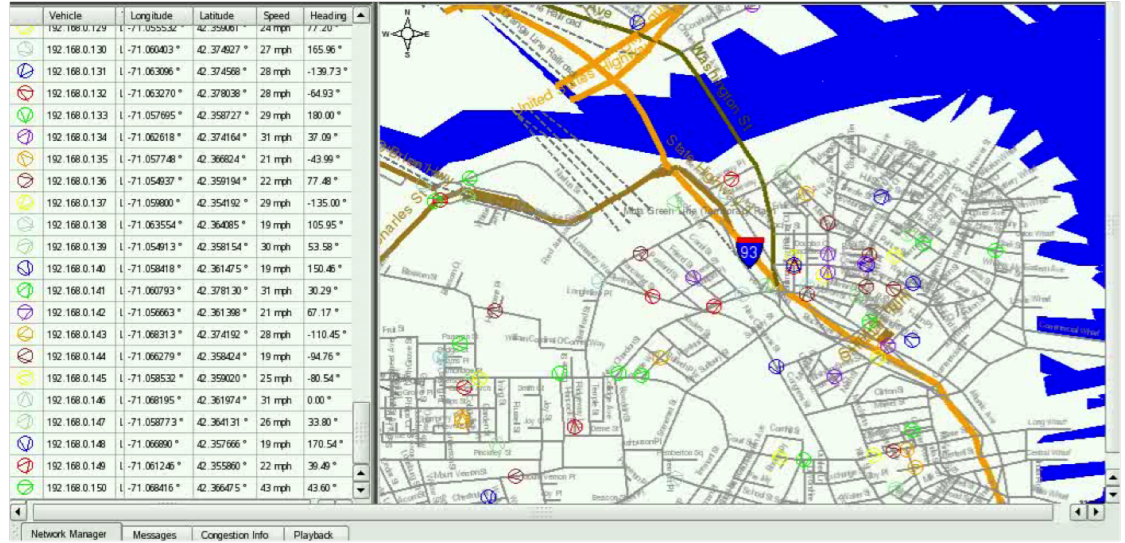


Figure 4.13: Groovenet. An example of GrooveNet simulation, where vehicles are moving in a real map with their coordinates being updated almost in realtime.

network model and a GUI based on Qt. GrooveNet can load real street maps from the TIGER/Line database in order to simulate vehicles mobility on real roads, including fixed mobility, street speed, uniform speed, and car-following mobility models. GrooveNet also supports many operational modes such as drive mode, simulation mode, playback mode, hybrid simulation mode, and test generation mode. Furthermore, the simulator has capabilities such as communicating with surrounding vehicles

¹<http://www.seas.upenn.edu/~rahulm/Research/GrooveNet/>

²RoadNav is an open source street navigation solution capable of running on a variety of operating systems. It can obtain your position from a GPS unit, plot a map of your area, and provide directions to locations in the USA. It can also verbalize directions using Microsoft SAPI 5.1, Festival, flite, and OS X's built in text to speech engine.

Roadnav uses the free TIGER/Line (Topologically Integrated Geographic Encoding and Referencing) files from the US Census Bureau to build the maps, along with the GNIS state and topical gazetteer data from the USGS to identify locations. It has experimental support for scripting, LCDproc, importing OpenStreetMap data, and importing GPX waypoints and tracks.

4. VEHICULAR NETWORK SIMULATORS

to get real traffic information, executing a specific scenario, mixing between the previous two modes, and finally playing back the logfile generated during any of the modes operation for further analysis. GrooveNets unique ability to integrate simulated vehicles with real vehicles allows it to function as testbed software as well as a simulator.

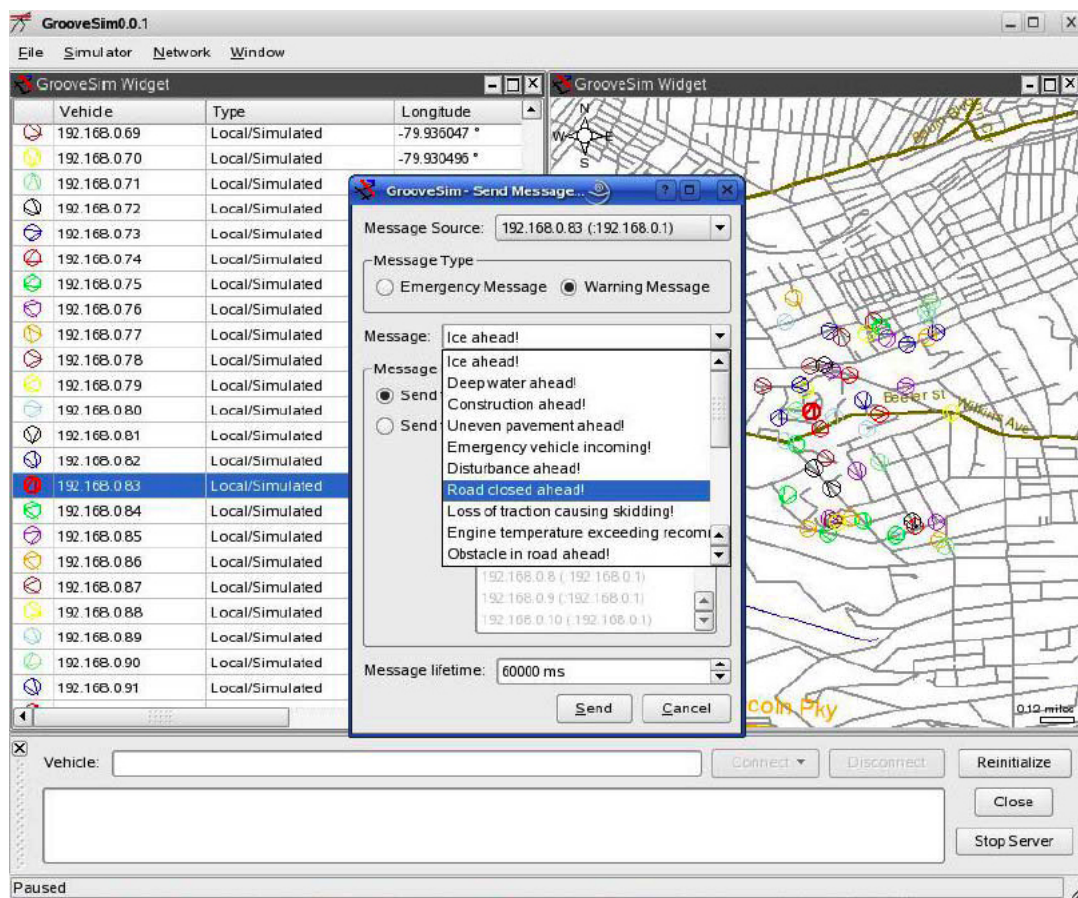


Figure 4.14: Groovenet. The GrooveNet simulator allows to broadcast emergency alerts or road warnings, thus analyzing the effect on the vehicles in realtime. This unique characteristic makes GrooveNet suitable both for simulations and real research testbeds.

4.4.3 TraNS

TraNS¹ (Traffic and Network Simulation Environment) (76) can be called the first VANET simulator. It was the first work to combine a network simulator, ns-2, with a

¹<http://trans.epfl.ch/>

vehicular mobility simulator, SUMO, and to provide feedback from the network simulator to the mobility simulator. TraNS can operate in two modes: *network-centric mode* and *application-centric mode*. In the network-centric mode, there is no feedback provided from ns-2 to SUMO, so the vehicles mobility trace file can be pre-generated and fed to the network simulator later. The link between the two simulators in this case is done through a parser that analyzes the mobility trace file generated by SUMO and converts it to a suitable format for ns-2.

In the application-centric mode, the feedback between ns-2 and SUMO is provided

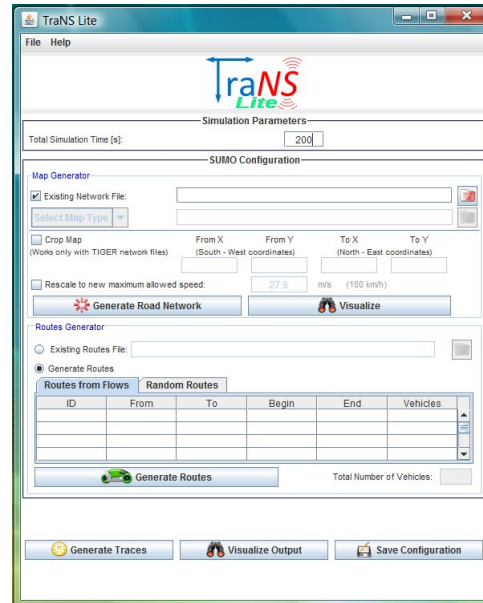


Figure 4.15: TraNS simulator. Screenshot of a light version of TraNS, called TraNS Lite.

through an interface called TraCI (100). In this mode the two simulators (SUMO and ns-2) must run simultaneously. TraCI achieves the link between ns-2 and SUMO by converting the mobility commands coming from ns-2 to a sequence of mobility primitive commands such as stop, change lane, change speed, and so on that can be sent to SUMO. As both simulators are running separately at the same time, the two-way communication in application-centric mode uses two separate events queues.

4. VEHICULAR NETWORK SIMULATORS

4.4.4 Veins

Veins (Vehicles in Network Simulation) (90) is another simulator that couples a mobility simulator with a network simulator: SUMO is paired with OMNeT++ by extending SUMO to allow it to communicate with OMNeT++ through a TCP connection. In order to create a bidirectional communication between the two simulators, OMNeT++ has also been extended by adding a module that allows all participating nodes (vehicles) to send commands via the established TCP connection to SUMO. In this case, the two extensions represent the interface between the network simulator and the mobility simulator. Thus, the network simulator can react to the received mobility trace from the mobility simulator by introducing new nodes, by deleting nodes that have reached their destination, and by moving nodes according to the instructions from the mobility simulator.

In Veins, there is a manager module that is responsible for synchronizing the two simulators. At regular intervals, the manager module triggers the execution of one timestep of the traffic simulation, receives the resulting mobility trace, and triggers position updates for all modules it had instantiated. Thus, as with TraNS, this simulator has two separate events queues.

4.4.5 NCTUns

NCTUns¹ (National Chiao Tung University Network Simulation) (99) implements two-way communication with a single events queue. NCTUns 1.0 was developed only as a network simulator, but the most recent version, NCTUns 6.0 (released on 15 January, 2010), integrates some traffic simulation capabilities, such as designing maps and controlling vehicles mobility. A large variety of maps can be designed using different types of supported road segments (e.g., single-lane roads, multilane roads, crossroads, T-shape roads, and lane-merging roads). Also, NCTUns includes a GUI to aid in the design process of the maps.

The supported vehicular movement has two modes, *prespecified* and *autopilot*. In the prespecified movement mode, the scenario designer specifies the moving path and the speed for each vehicle. In autopilot mode, the scenario designer specifies the following

¹<http://nsl10.csie.nctu.edu.tw/>

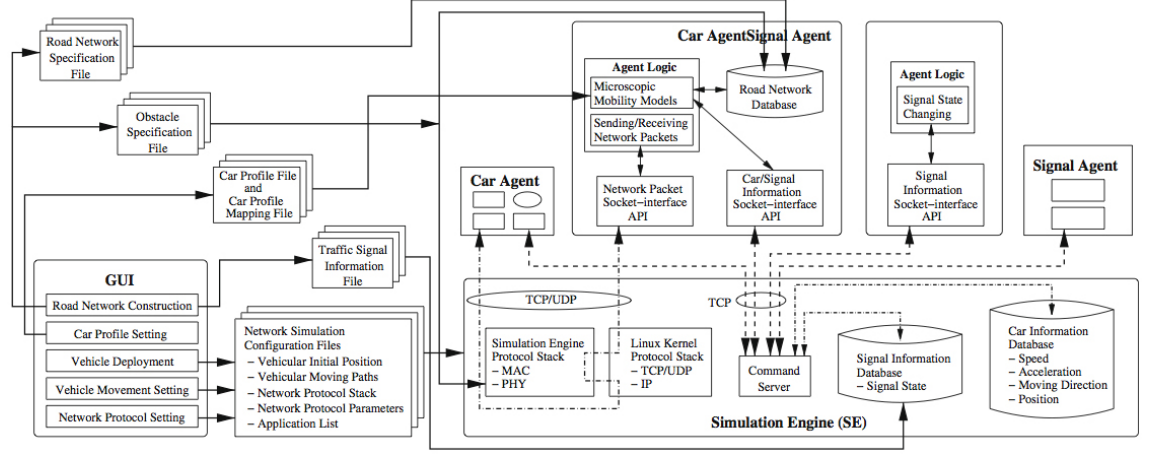


Figure 4.16: NCTUns simulator. The architecture of NCTUns 6.0 network simulator.

parameters for each vehicle: initial speed, maximum speed, initial acceleration, maximum acceleration, maximum deceleration, and so on. Then, the autopilot selects the best route to navigate in the map. Autopilot mode is also capable of performing car following, lane changing, overtaking, turning, and traffic light obeying. The best feature available in NCTUns is that its network protocol stacks includes the Linux kernel protocol stack, including TCP/IP and UDP/IP, and the user level protocol stack and the MAC and PHY layer protocols. *This means that it is possible to run on the simulated nodes whatever application which runs on Linux!* Furthermore, as the simulator works directly on the Linux kernel's network stack, it is not needed the protocol validation (the first version of NCTUns did not rely on real network stack, thus requiring a validation). However, the main disadvantage of NCTUns is that the code for the vehicles movement logic is integrated with the network simulation code, which makes it difficult to extend. Another disadvantage of this simulator is that, at the moment, it is released for one Linux distribution, Fedora 12 (as it is based on the linux kernel, the portability suffers a bit).

4.4.6 ASH

ASH (Application-aware SWANS with Highway mobility) (7) is an extension of the wireless network simulator SWANS that implements the IDM vehicular mobility model and MOBIL lane changing. ASH supports feedback between the vehicular mobility sub-

4. VEHICULAR NETWORK SIMULATORS

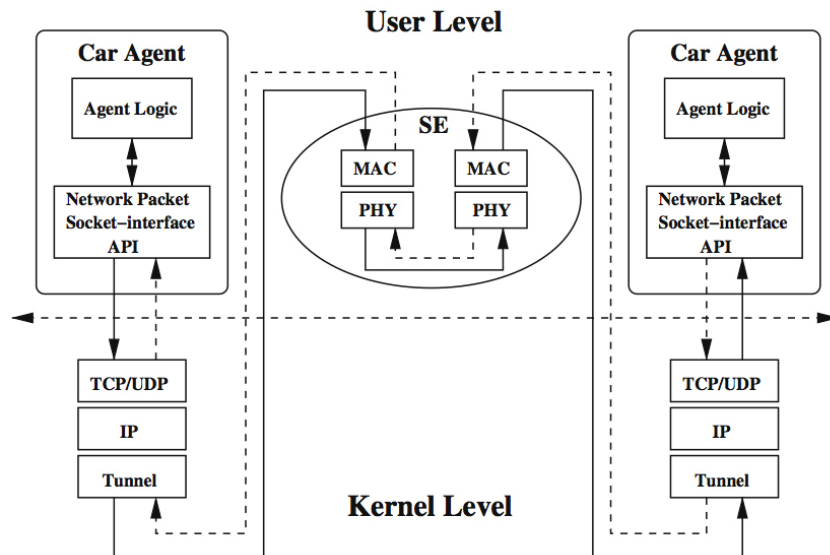


Figure 4.17: NCTUns simulator. The architecture of NCTUns 6.0 network protocol simulation.

simulator and the network subsimulator, making it one of the two-way communication simulators with a single events queue. ASH allows users to design a simple highway segment and customize it by specifying the directions (one-way or two-way), the number of lanes, the number of entries and exits and their corresponding locations along the segment.

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

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

4.4 Tightly Integrated Simulators

Attribute	SUMO/Move TraNS	VanetMobiSim	NCTuns
<i>Custom graphs</i>	supported	supported	supported
<i>Random graphs</i>	grid based	voronoi graph	shape-file
<i>Map based graphs</i>	Tiger DB	GDF	bitmap image
<i>Multilane graphs</i>	supported	supported	supported
<i>Start/End position</i>	AP, random	AP, random	random
<i>Trip</i>	random start-end	random start-end	random
<i>Path</i>	random walk, Dijkstra	random walk, Dijkstra	random walk
<i>Velocity</i>	road dependent smooth	road dependent smooth	road dependent smooth

Table 4.1: Traffic level features in SUMO/Move/TraNS, VanetMobiSim and NCTuns mobility simulators.

Attribute	SUMO/Move/TraNS	VanetMobiSim	NCTuns
<i>Human patterns</i>	car following models	IDM, IDM-IM IDM-LC	IDM, IDM-IM IDM-LC
<i>Intersection management</i>	Stoch turn	traffic lights and signs	traffic lights
<i>Lane changing</i>	not supported	supported	supported
<i>Radio obstacles</i>	not supported	supported	supported

Table 4.2: Motion level features in SUMO/Move/TraNS, VanetMobiSim and NCTuns mobility simulators.

4. *static silent node*: this represents a road obstacle that should execute a null application.

In particular, the addition of the mobile silent node is important to allow testing of protocols under different penetration rates, where not all vehicles are equipped with communication devices. The location of the static silent node can either be predetermined before running the simulation or can be determined at runtime, in order to simulate an accident, for example.

Because most VANET applications use flooding-based techniques to disseminate data, ASH also implements the Inter-Vehicle Geocast protocol (IVG) (12). Moreover, it supports a probabilistic version of IVG to take the surrounding traffic density into account.

4. VEHICULAR NETWORK SIMULATORS

Attribute	SUMO/Move/TraNS	VanetMobiSim	NCTuns
<i>GUI</i>	not supported	supported	supported
<i>Output</i>	ns-2, GlomoSim Qualnet	ns-2, GlomoSim Qualnet	ns-2
<i>Level of integration</i>	federated/integrated	separated	integrated

Table 4.3: Other features in SUMO/Move/TraNS, VanetMobiSim and NCTuns mobility simulators.

ASH also implements statistical and logging utilities to support simulations. The utilities provide information about the simulation entities at different granularity. This information can be retrieved at the simulation level, lane level, vehicle level, or message type level. The statistics utility provides statistics about all possible events that can occur in the simulation.

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

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

Figure 4.18: Benefits and drawbacks. Benefits and drawbacks of several VANETs simulator.

4.5 Scalability of VANET simulators

This section aims to provide a brief overview of network simulators scalability, with an experimental result on Qualnet and Ns2, two simulators which have recently gained attention in the research community. When simulating a complex scenario including vanet networks, with both mobility and signal transmission and propagation models, it

might be valuable to understand how far you can go with your simulation. This means that the most important parameters that every researcher should keep in mind before approaching any kind of simulation are *scalability* and *overall performance*. Scalability, which is, generally speaking, the ability to either handle growing amounts of work in a graceful manner or to be readily enlarged, gives to the researcher an idea on the maximum scenario he is allowed to simulate, in terms of number of mobile nodes, terrain dimensions, number of simultaneous protocols and details' level of the simulation's outcome. For example, one may simply think to simulate a large network of thousands of nodes with many details per network layer but if the simulation time or the memory consumption start growing too fast, this may turn out to be unfeasible.

Trying to compare two network simulators is not an easy task to accomplish, first of all because of the different implementations of both kernel and protocols, which may result in completely different resource consumption. Secondly, it should also be considered the detail's level of the simulation itself. In order to gain some experimental results of the performance trend in Qualnet and Ns2 (for a quick comparison), we implemented in both the simulators the following simulation set-up:

- *simulation time*: 600 secs;
- *simulation area*: 2000x2000 sqm.;
- *number of mobile nodes*: up to 1000 in Ns2, up to 4000 in Qualnet;
- *mobility*: random waypoint with initial uniform distribution;
- *wireless protocol*: 802.11b with 150 m nominal radio range, two-ray propagation pathloss model and constant shadowing model;
- *routing protocol*: AODV
- *traffic*: a couple of nodes always involved in UDP CBR ("*Constant Bit Rate*") transfer with a packet size of 1460 bytes.

The difference in the range of mobile nodes is due to the limitation in the Ns2 scalability which, compared to Qualnet, does not allow simulation of thousands of nodes. In fact, in (110) it was proved that Ns2 does not scale well for wireless sensor network (and then it makes sense to extend the results for Vanet, which is a specialization of a

4. VEHICULAR NETWORK SIMULATORS

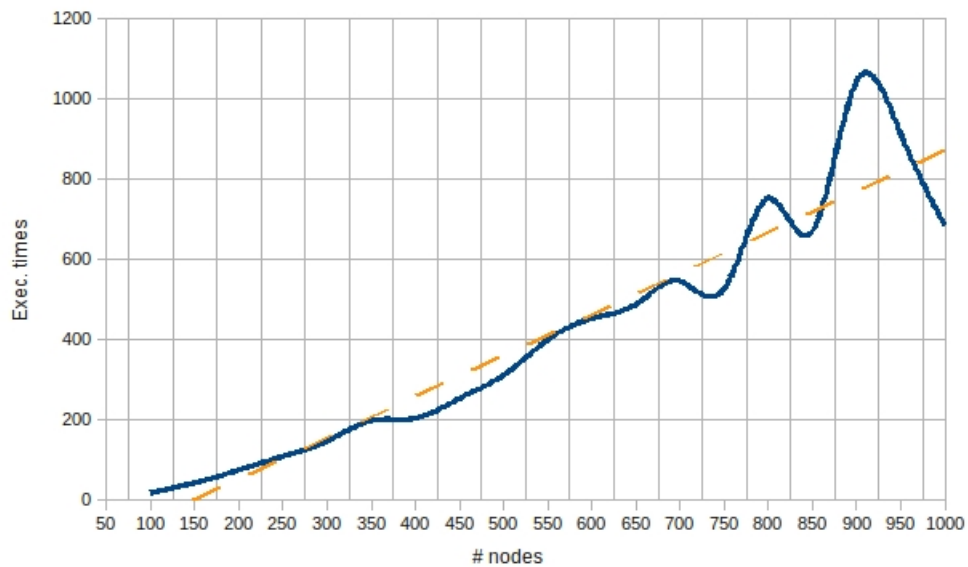


Figure 4.19: Simulation time in Ns2. Simulation time in Ns2 (version 2.34) with mobile nodes ranging from 50 up to 1000 in a 2000x2000 sqm area.

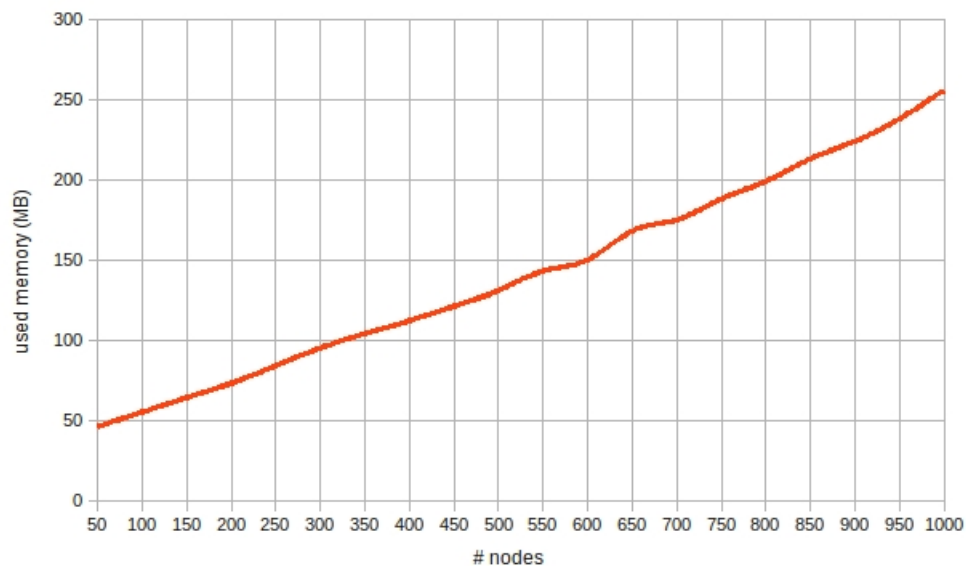


Figure 4.20: Memory memory consumption in Ns2. Memory consumption in Ns2 (version 2.34) with mobile nodes ranging from 50 up to 1000 in a 2000x2000 sqm area.

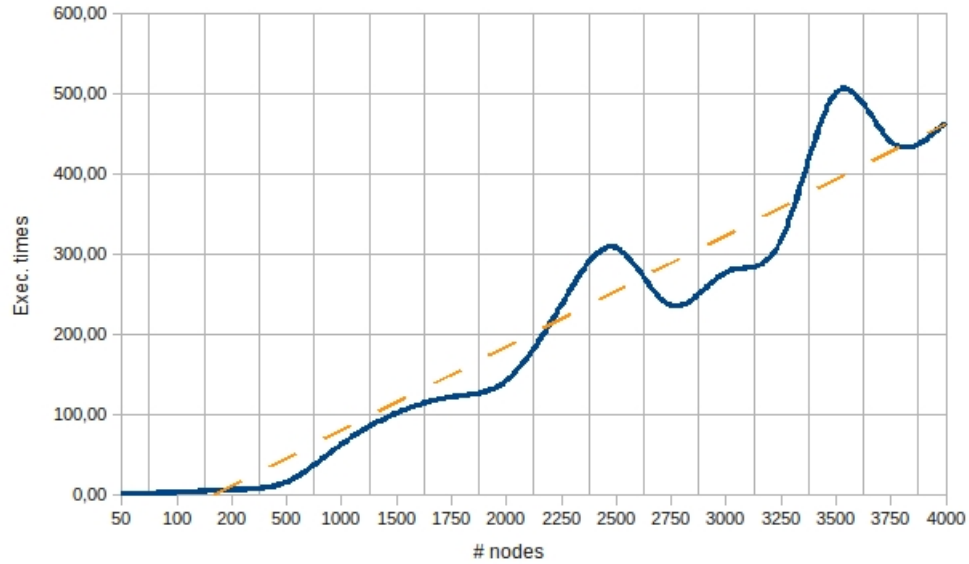


Figure 4.21: Simulation time in Qualnet. Simulation time in Qualnet (version 4.5) with mobile nodes ranging from 50 up to 4000 in a 2000x2000 sqm area.

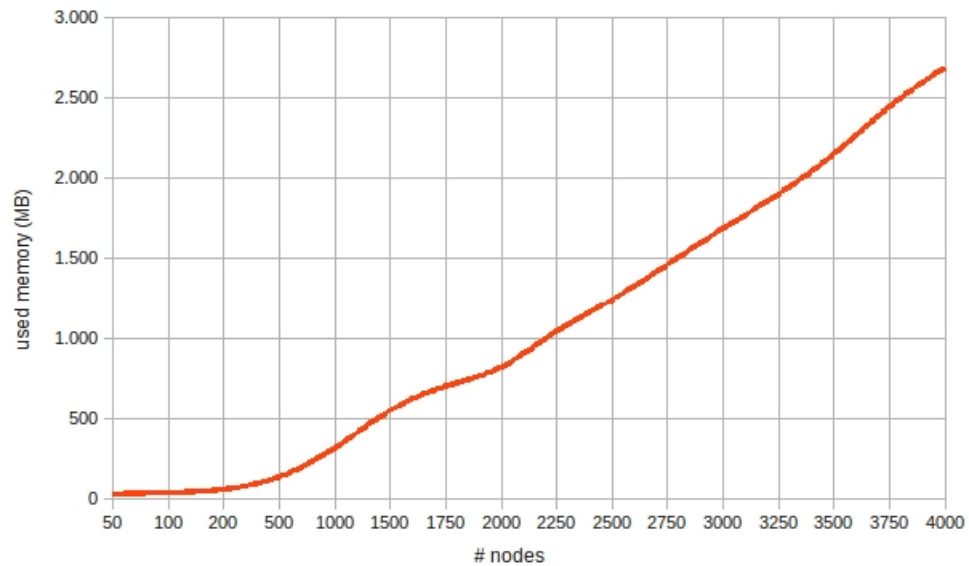


Figure 4.22: Memory memory consumption in Qualnet. Memory consumption in Qualnet (version 4.5) with mobile nodes ranging from 50 up to 4000 in a 2000x2000 sqm area.

4. VEHICULAR NETWORK SIMULATORS

sensor network).

The results of the simulations (figures: 4.19, 4.20, 4.21 and 4.22) show clearly that the performance trend of both Ns2 and Qualnet is almost linear. The flatters on the both graphs is mainly related to the increasing number of nodes in a relatively small simulation area (as the nodes, randomly moving in the area, come too close each other, the high number of collisions can raise up the number of the events the simulator has to handle). Although the performance trend in Ns2 and Qualnet is almost linear, anyway it should be noticed the big difference in terms of absolute values: if considering the trend line, Ns2 needs almost 900 seconds to perform the same simulation that Qualnet handles in 63 seconds and this is primarily due to the different implementations of the core functions. It should also be highlighted that Qualnet execution times are shorter even it performs more accurate simulations at physical and mac layers.¹

¹SNT whitepapers prove Qualnet precision and detailed level of simulation in their whitepapers (they cannot be referenced in this site). For further information, please refer to the SNT website at: <http://www.scalable-networks.com/>.

Chapter 5

Data dissemination survey

This chapter will survey the different types of information exchange adopted in vehicular networks with common practices and methodologies that have been considered in research literature (e.g.: *opportunistic exchange of resources between vehicles, vehicle assisted data delivery, cooperating downloading of information, etc.*) with a special emphasis to the network coding technique.

5.1 Types of information involved in the dissemination process

The different types of information that need to be communicated and then shared in a vehicular network can be roughly classified in four categories:

- *Safety information*

Safety information is probably the most important information type that are communicated in a vehicular network (109), as motor vehicle crashes cause more than 120.000 victims in Europe and 1.2 million worldwide¹. As you may recall, in chapter 2 we have introduced DSRC (Dedicated Short-Range Communications), a variant of the 802.11a technology designed to support V2V and V2I and conceived by the Federal Communications Commission (FCC) to *"increase traveller safety, reduce fuel consumption and pollution and continue to advance*

¹Fact Sheet EURO/03/04 (Copenhagen, Rome, 6th April 2004) of the World Health Organization, ONU.

5. DATA DISSEMINATION SURVEY

the nation's economy". DSRC will support safety-critical communications such as collision warnings as well as other valuable Intelligent Transportation System applications (i.e.: electronic-toll-collection (ETC), realtime traffic advisories, digital map updates, etc.). As it concerns safety messages, FCC recognizes *safety messages* and *safety-of-life messages*, which have the highest priority, whether originated by vehicles or roadside transmitters. Thus, DSRC can be used to prevent collisions between vehicles by providing important informations to the drivers about whether the vehicle ahead is braking, if the speed is too high or the distance to other vehicles (or objects, more in general) is getting too close. It is valuable to remark that the US Department of Transportation and the Vehicle Safety Communications Consortium (VSCC) have identified eight applications that rely on security informations(81):

1. *Curve Speed Warning*

The Curve speed warning systems use a combination of GPS and digital maps to assess threat levels for a driver approaching a curve too quickly: if the driver enters a curve at a speed that will not allow him to drive through the curve safely, he will automatically receive a warning. This warning can be an acoustic signal or a visual symbol that appears in the navigation system or directly on the instrument panel.

2. *Traffic Signal Violation Warning*

The traffic signal violation warning system is designed to warn a vehicle when the system detects that the vehicle is in danger of running the traffic signal. The system will make a decision (sending a warning message to the vehicle) based on the traffic signal status and timing and the vehicles speed and position. Some other factors are considered in such situations, such as the road surface and weather conditions.

3. *Emergency Electronic Brake Lights*

This application alerts the driver when a preceding vehicle performs a severe braking maneuver by trasnmitting the alert notification with DSRC. Surrounding nodes that receive the message must then discern if the event is relevant, since, as we have already said, they can ignore warnings from vehicles traveling behind, far ahead, or in the opposite direction.

5.1 Types of information involved in the dissemination process

4. *Pre-Crash Warning*

This system is designed to reduce the severity of an accident. Depending on the system, they may warn the driver, precharge the brakes, retract the seat belts removing excess slack and automatically apply partial or full braking to minimize the crash severity.

5. *Left Turn Assistant*

The left turn assistant system helps a driver to make a safe left turn at signalized intersections with no left turn arrow phase. The system will collect data about vehicles coming from the opposite direction that are approaching the intersection, using sensors located at the intersection or DSRC communications, and then inform vehicles that are going to make the left turn. One way to implement such system is by having data collected continuously, and when there is a vehicle with its left turn signal on, the system will send a message to that vehicle about the traffic traveling in the opposite direction of the vehicle. The other way is to have an in-vehicle system that sends a request to be notified about the traffic in the opposite direction when the left turn signal is activated. The system will then collect data about the traffic in the opposite direction and send a message to the vehicle that requested the information. In both ways, the driver will be informed and warned about the traffic coming in the opposite direction. The system is different to most of the other intersection collision avoidance systems because it can use both V2I and I2V communications.

6. *Lane change Warning*

These in-vehicle electronic systems monitor the position of a vehicle within a roadway lane and warn a driver if it is unsafe to change lanes or merge into a line of traffic. These systems are rearward-looking and assist drivers who are intentionally changing lanes by detecting vehicles in the driver's blind spot.

7. *Stop Sign Movements Assistance*

The stop sign movement assistant system is designed to avoid accidents at stop sign intersections. The system uses data collected by sensors or DSRC

5. DATA DISSEMINATION SURVEY

communications to inform the driver if it is unsafe to go through an intersection, so it warns drivers if there is traffic coming through the intersection at the same time. This system uses both V2I and I2V communications.

8. *Cooperative Forward Collision Warning*

This warning system is designed to warn drivers that are approaching the same intersection if there is a possibility of an accident occurrence. Data about all vehicles approaching an intersection are gathered using sensors and/or DSRC communications and then processed to see if there is a chance of a collision at the intersection. If there is such a situation, messages will be sent to vehicles that might be involved in this unsafe situation. The data required by the system includes information regarding the vehicles that are approaching the intersection and other data about the road. The data needed about vehicles are position, velocity, acceleration, and turning status. The data needed about the road are shape and surface. Data about the weather status is helpful in calculating the chance of a collision occurrence.

The communication requirements of these eight applications are shown in table 5.1. Note that communication frequency ranges from 1-50 Hz, the size of the packet ranges 200-500 bytes, and the maximum communication range spans from 50-300 meters. Further, high-level Data Element requirements are specified and several of these Data Elements (e.g. position and heading) are needed by multiple applications. Responding to these identified applications, the "Dedicated short range message set" defines over seventy vehicle Data Elements (e.g. heading, acceleration (with varying precision: 4bit, 8bit, 16 bit), headlight status and brake status). Of these Data Elements, thirty of the most commonly used elements are selected for a common message set, as listed in table 5.2. However, very few of these applications have actually been implemented or fully developed. Hence, the exact usage characteristics of the safety messages were in flux at the time the safety messages were being defined and standardized.

Clearly, in each of the said applications, the time taken by the driver of a motor vehicle to react to warnings has to be considered while delivering information(109), thus introducing the human component into the picture as well as the traffic density and the environment, factors which influence the delivery of safety messages.

5.1 Types of information involved in the dissemination process

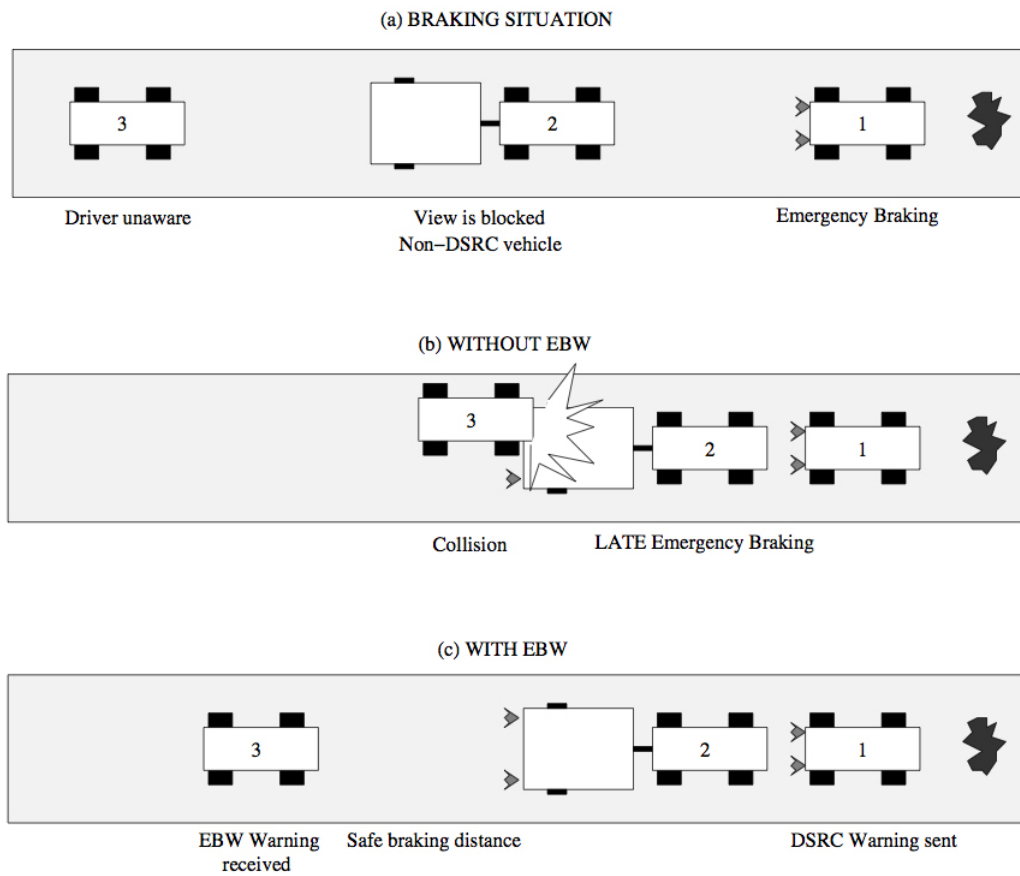


Figure 5.1: EBW - Emergency Brake Warning. The EBW alerts the driver behind by augmenting the brake-lights notification system when a preceeding vehicle performs a severe braking maneuver.

5. DATA DISSEMINATION SURVEY

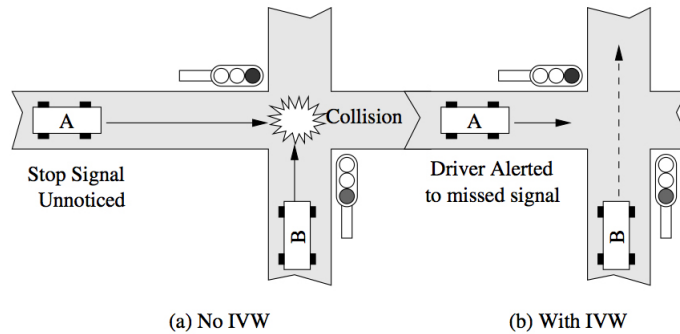


Figure 5.2: IVW - Intersection Violation Warning. The IVW warns the driver if violating a red light seems imminent. Other vehicles approaching the traffic lights are also warned that a close vehicle issued a warning message.

- *Traffic information*

Congestion of roads is a very important information to share on vehicular networks in order to let personal electronic navigators to select the best route and avoid traffic. This can be extremely useful in case of car crash or even under emergency circumstances because mobile nodes can be informed in advance on the congested area so that service vehicles can quickly reach the destination. Of course, this kind of messages are without any doubt useful for safety and efficiency but less time-critical than the "life safety" ones discussed earlier. With respect to infotainment and content information, this class of data still has higher priority.

- *Infotainment information*

With infotainment information we intend all that kind of messages about entertainment and, more in general, useful messages to the driver. An example can be:

- locating the closest coffee shop
- locating the closest cinema or mall
- locating the fuel station which offers the best price in that area
- locating the closest available parking spot
- many and many others...

5.1 Types of information involved in the dissemination process

Application	Comm. type	Freq.	Latency	Data transmitted	Range
traffic signal violation	I2V one-way, P2M	10 Hz	100 ms	signal status, timing, surface heading, light pos., weather	250 m
curve speed warning	I2V one-way, P2M	1 Hz	1000 ms	curve location, curvature, bank speed limit, surface	200 m
emergency brake lights	V2V two-way, P2M	10 Hz	100 ms	position, deceleration heading, velocity	200 m
pre-crash sensing	V2V two-way, P2P	50 Hz	20 ms	vehicle type, yaw rate, position heading, acceleration	50 m
collision warning	V2V one-way, P2M	10 Hz	100 ms	vehicle type, position, yaw rate heading, velocity, acceleration	150 m
left turn assist	I2V and V2I one-way, P2M	10 Hz	100 ms	signal status, timing, position direction, road geom., heading	300 m
lane change warning	V2V one-way, P2M	10 Hz	100 ms	position, heading, velocity acceleration, turn sign. status	150 m
stop sign assist	I2V and V2I one-way	10 Hz	100 ms	position, velocity heading, warning	300 m

Table 5.1: Eight high priority vehicular safety applications as chosen by NHTSA (71) and VSCC (*Vehicle Safety Communications Consortium*), which is a consortium born in May, 2002 with the mission to facilitate the advancement of vehicle safety through communication technologies (the members are: BMW, DaimlerChrysler, Ford, GM, Nissan, Toyota, and VWGM, VW). Communication frequencies ranges from 1-50 Hz and the maximum range span from 50-300 meters. Note that P2M represents "point-to-multipoint", I2V "infrastructure-to-vehicle" and V2I "vehicle-to-infrastructure".

- *Content*

This kind of information differs from the others especially in the quantity of data exchanged between nodes and recent research works in vehicular networks have focused on content distribution as multimedia data are becoming essential in everyone's life(55)(68)(28); this means the main assumption is that vehicles are interested in obtaining extremely large files (e.g. videos and movie clips) that cannot be shared through a limited transactions with either the infrastructure or the nearby mobile nodes. The discovery and distribution of large quantities of information is still a challenging problem especially in a dynamic environment

5. DATA DISSEMINATION SURVEY

such as a vehicular network.

5.2 Data dissemination techniques

The process of information dissemination determines a bunch of problems because of many reasons. At first glance, we have to consider the network size but also the vehicles' speed and the patchy and intermittent connectivity between mobile nodes; moreover, there is another problem which can severely affect the entire process: *latency requirements*(109). As a consequence, content information has to be quickly discovered and shared among nodes.

In the research literature, many methods of information delivery are presented and, generally speaking, we can distinguish the following data dissemination approaches:

- *opportunistic*: information is pulled from other vehicles or the infrastructure as a target vehicle encounters them(108);
- *vehicle-assisted*: a vehicle carries information with it and delivers it either to the infrastructure or to other vehicles when it encounters them. This process involves mobility in addition to wireless transmissions in order to disseminate the information(113);
- *cooperative*: vehicles can download partial units of some content and then share them afterwards to obtain the complete content. This method is particularly suitable for content dissemination (where the amount of information is rather important in terms of file size) and it was adopted to develop our dissemination protocol based on rateless codes(68) (discussed in detail in chapter 6).

In addition to these dissemination methods, it is important to take into consideration the *spatial and temporal constraints*(108)(105), that is evaluating how far the information should be disseminated(109). For example, in case of traffic and service information, aging with time and distance is suggested in (108), where data, opportunistically pulled from neighboring nodes, are "marked" with a time stamp and a location stamp. As the mobile node moves farther away from where the information is relevant (or as time elapses), the information itself is aged or eventually purged, thus allowing vehicles to maintain up-to-date information without penalizing memory

or other resources. Because of the spatio-temporal relevance of information, a piece of information tends not to propagate beyond a specific boundary; in time, a given piece of information would propagate very quickly till it reaches a maximum number of copies and then it would also rapidly decline from that point.

As it concerns the content distribution in vanets, several solutions have been discussed in the last few years. The basic idea is that vehicles download fragments of files when they have access to the roadside and then share them with other mobile nodes on the way. Although it seems a simple approach, there are many issues related to the sporadic nature of the connectivity.

Sometimes, the content is partially downloaded from a roadside base station, which also provides a list of other vehicles that have parts of that file, like the solution proposed by Nandan et al. in (68). On the contrary, in the work we presented at WCNC'09 (20), the resource is entirely downloaded by the infrastructure (or other vehicles) and then shared among mobile nodes without any provided assistance or list of any kind by the base station. In the first scenario, each vehicle "gossips" about what part of the content it has, while in the second, it answers to a specific request of other nodes.

In *BitTorrent* (68), a closest-rarest strategy is adopted where the rarest piece of the file is downloaded first, but it is decentralized and employs proximity information in order to improve the overall performance. This is because, in general, multihop communications degrade the performance of the information delivery rapidly as the number of hops increases. To overcome the said problem, we can cite the work done in *CodeTorrent* (cfr. 5.4.2) (55), where *network coding technique* and *mobility assistance* along with gossiping (as in (68)) can be used with one single hop distribution of content, thus further improving performance.

5.3 Information sharing via Network Coding technique

5.3.1 Introduction to Network Coding

In scenarios where delay-tolerant applications are implemented, different approaches can be applied in order to reduce as much as possible the number of transmissions required to deliver some amount of data, thus improving the overall performance. This goal is especially interesting in VANETs¹, where the reduction of the traffic plays an

¹More in general, it is also relevant in pervasive systems.

5. DATA DISSEMINATION SURVEY

important role in guaranteeing better performance. To achieve this aim we can adopt recent network paradigms which introduce in the network protocols some kind of intelligence to further increase the efficiency in data transmissions. Data dissemination is one of the most important services in such networks: it can be used to disseminate control data such as routing information, device status, performance metrics and, more in general, any kind of interesting content. Furthermore, data broadcasting could also be useful when a lot of users want to share some kind of data in a distributed way: downloading the same files, sharing knowledge about the network services and so on (as we saw in *content data dissemination*). In all these scenarios, similar information, usually distributed over the whole network, has to be exchanged by the users.

The heterogeneity of the devices together with the wireless characteristics of the networks, make the definition of an efficient data dissemination scheme a challenging issue: the main goal here is to guarantee a limited usage of the network resources in order to increase the throughput. This translates in defining a data dissemination scheme which is able to deliver a lot of data at the minimum cost in terms of number of transmissions. We know that the transmission phase is particularly expensive for wireless devices and, in addition, high traffic leads to unsatisfactory performance due to the contention mechanisms used in wireless communications. Thus, reducing the number of transmissions to deliver data to multiple users is a very important goal in such networks. Furthermore, if we think of VANETs, the reduction of transmission is essential because of the mobility: we could say that the communication has to be "*effective*".

A lot of studies, in literature, follow this direction by proposing different solutions. Recently, a technique known as *in-network data aggregation* has been developed for wireless sensor networks (26). It consists of some data processing along the network in order to combine packets and reduce the amount of information to be spread. However, since the data processing performed over the packets is generally lossy, in-network data aggregation is useful when nodes are interested on some average measures of the required information. For instance, it is particularly useful in wireless sensor networks for environmental monitoring to gather some information about the average temperature in a specific area and so on. This kind of approach, however, can not be applied for lossless data dissemination services, such as those generally related to the VANETs. However, the in-network data aggregation approach suggests that for an efficient data dissemination we have to go to a specific direction: introducing some intelligence at

the network layer or even at application layer (as we did in our protocol (20)) in order to carry more information by transmitting the same number of packets, in some how coded .

The Network Coding technique is one of the promising approach because it allows to achieve the said goal without loss of information. In the following paragraphs we study how to apply network coding paradigm in data dissemination in order to make the data delivery more efficient in terms of energy and throughput.

5.3.2 What is Network Coding

Network coding is a new research area that can offer interesting applications in practical networking systems. With network coding, intermediate nodes may send out packets that are linear combinations of previously received information. There are two main benefits of this approach: potential throughput improvements and a high degree of robustness. Robustness translates into loss resilience and facilitates the design of simple distributed algorithms that perform well, even if decisions are based only on partial information.

The concept of network coding is firstly introduced in (9) by Ahlswede et al. and it can be formulated in various ways at different levels of generality. A simple definition is reported next:

"Network coding is a particular in-network data processing technique that exploits the characteristics of the wireless medium (in particular, the broadcast communication channel) in order to increase the throughput of the network."

In other words, it is a packet dissemination strategy which can be used to improve the throughput, thus guaranteeing high performance. In contrast to the store and forward paradigm, network coding implements a more complex *store, code, and forward* approach where each node stores the incoming packets in its own buffer, and, at transmission time, sends a combination of the stored data. To successfully decode, say, K packets, a node must collect K independent combinations. It can provide the highest gains in multicast or broadcast networks. More specifically, network coding can typically achieve higher transmission rates than separate unicast transmissions when

5. DATA DISSEMINATION SURVEY

information sources transmit to multiple destinations or to all nodes in the network. The basic mechanism of network coding is introduced in (9) through a simple example which we report for readers ease: consider the acyclic network depicted in Fig.5.3, with 6 nodes and a single source S . Time is subdivided in slots in a TDMA fashion and wireless links can carry a single flow at a time, i.e., either flow $b1$ or $b2$ can be transmitted during a time slot over a specific link. The problem to be solved is to send two bit flows $b1$ and $b2$ to both destinations Y and Z by exploiting the network at its maximum capacity. In other words, the flows are to be multicast to both destinations in the most efficient way.

As a first attempt to solve the problem, one might devise a simple transmission schedule which consists of alternating two distinct transmission modes between even and odd slots as follows. In even slots we may let links $T - X, X - Y, T - Y, X - Z$ carry flow $b1$, and links $S2 - U$ and $U - Z$ carry flow $b2$. Hence, during even slots node Y receives flow $b1$ only, whereas node Z receives both flows.

This transmission schedule is depicted in Fig.5.3.A. During odd slots, instead, the situation may be reverted according to the transmission schedule in Fig.5.3.B. In such a case, node Y receives both flows and node Z receives flow $b2$ only. Overall, this strategy leads to an average throughput of 1.5 flows per slot. A different way to attack the problem is provided by network coding and is shown in Fig.5.3.B. This time, node X derives from flows $b1$ and $b2$ the exclusive-OR $b1 \oplus b2$, which is finally passed to both destinations Y and Z . Destination nodes can now decode both flows by re-applying the XOR operation. This strategy obtains a throughput of 2 flows per slot, which is the maximum achievable in the above problem. We observe that the exclusive-OR is a simple form of coding. If the same objective is to be achieved by store and forward techniques (simple replication of the incoming data flow), then at least two subsequent slots are needed to deliver $b1$ and $b2$ to both destinations. With coding, the two flows can instead be delivered to both Y and Z in a single slot. It is therefore apparent that coding, besides offering advantages in terms of throughput, may also decrease the latency. Note, however, that a drawback of the above example is that the coding/decoding scheme has to be agreed upon beforehand. That is, node X must know in advance that it has to process the received flows by means of some coding strategy. While this is acceptable to present the advantages of coding by means of the above example, it has profound implications on the actual applicability of such

5.3 Information sharing via Network Coding technique

a technique in distributed and therefore uncoordinated networks (these implications together with possible solutions will be explored in the next paragraph).

The Network Coding technique offers another important benefit: *the increase of the information capacity, which is the capacity of the link (between two nodes) to transport innovative information.*

Referring to our above example, channel $X - Y$ and channel $X - Z$ have a transport capacity of a single flow per time slot. However, if we transmit a coded information over such a link, then *we obtain the higher transmission rate of 2 data flows per time slot.* In other words, the information capacity with coding is higher¹.

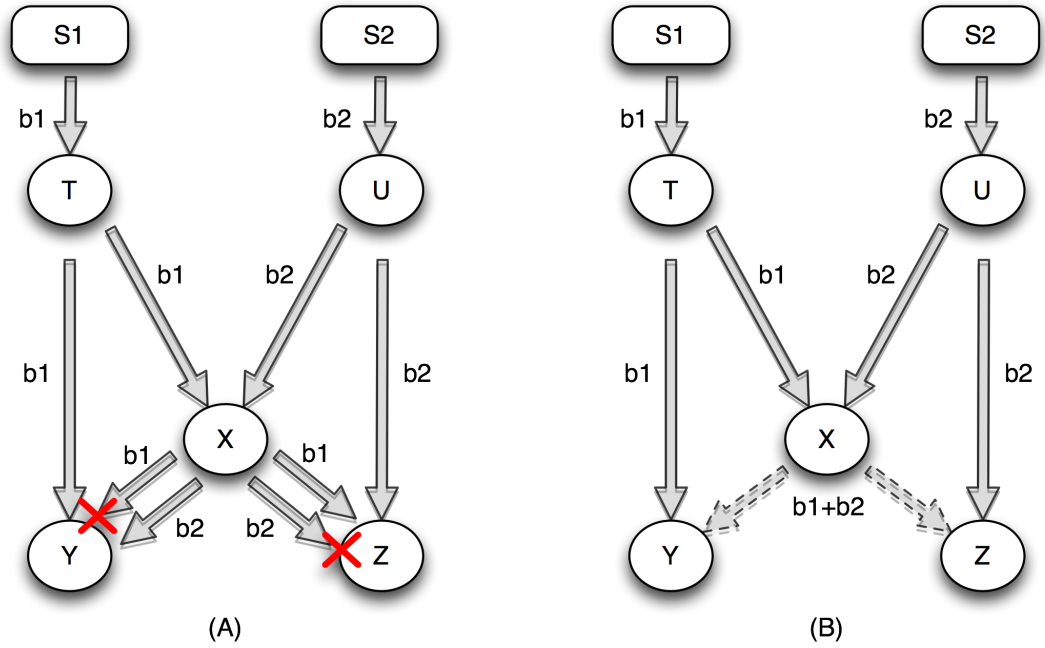


Figure 5.3: Network Coding. Simple example of Network Coding mechanism.

To better explain this idea, please consider the additional example reported in Fig. 5.4, which is closer to the wireless ad hoc communications. It represents a two hops network where each of *nodeA* (NA) and *nodeB* (NB) have to exchange a data packet, $b1$ and $b2$, respectively. Both the routing paths, namely the one from NA to NB and the other way around, go through *NodeC* (NC) which has to forward two packets.

¹This concept is extremely important, as it will be considered again in our protocol, VDRP.

5. DATA DISSEMINATION SURVEY

The first transmission phase, i.e., both NA and NB send their own packets to NC , which is equivalent to the classical communication paradigm and to network coding (see Fig. 5.4-b). Then, according to the classical communication paradigm, we need other two transmissions to deliver both $b1$ and $b2$ to the corresponding destination (see Fig. 5.4-c). With network coding, instead, we can use only one additional transmission by allowing NC to transmit the XORed version of packet $b1$ and $b2$ to both NA and NB as reported in Fig. 5.4-d. At the reception of this packet, NA (or NB) can decode packet $b2$ (or $b1$) by subtracting its own packet from the received one. This example shows how network coding is particularly effective whenever there are overlapping data flows, because it can exploit both the broadcast nature of the channel and the coding process to simultaneously deliver different packets to multiple users. After having briefly discussed the possible advantages offered by network coding, we now need to make things practical. In fact, it is actually infeasible to have pre-defined coding rules for every node in the network, as this would require full network knowledge. More than this, in distributed ad hoc scenarios, we need to cope with many constraints. For instance, nodes are not synchronized and, as a consequence, coding operation should not depend on time synchronization. Also, as messages travels through the network, they are exposed to delays (mainly due to processing at intermediate nodes and channel contention mechanisms). Finally, too complex coding schemes are actually to be avoided as they might be infeasible for resource limited wireless devices. Possible solutions to these issues are discussed in the following paragraphs.

5.3.3 Linear Network Coding

Network codes that involve only linear mappings are of particular interest as they can be executed at low computational cost. The word "*linear*" clearly let us imagine that the output flow of a given node is obtained as linear combination of its input flows (29). The coefficient of the combination are, by definition, selected from finite Galois field¹ and the information traversing a non source node has the following property:

"The content of any information flowing out of a set of non source nodes can be derived from the accumulated information that has flown into the set of nodes"

¹In abstract algebra, a finite field or Galois field (so named in honor of Evariste Galois) is a field that contains only finitely many elements.

5.3 Information sharing via Network Coding technique

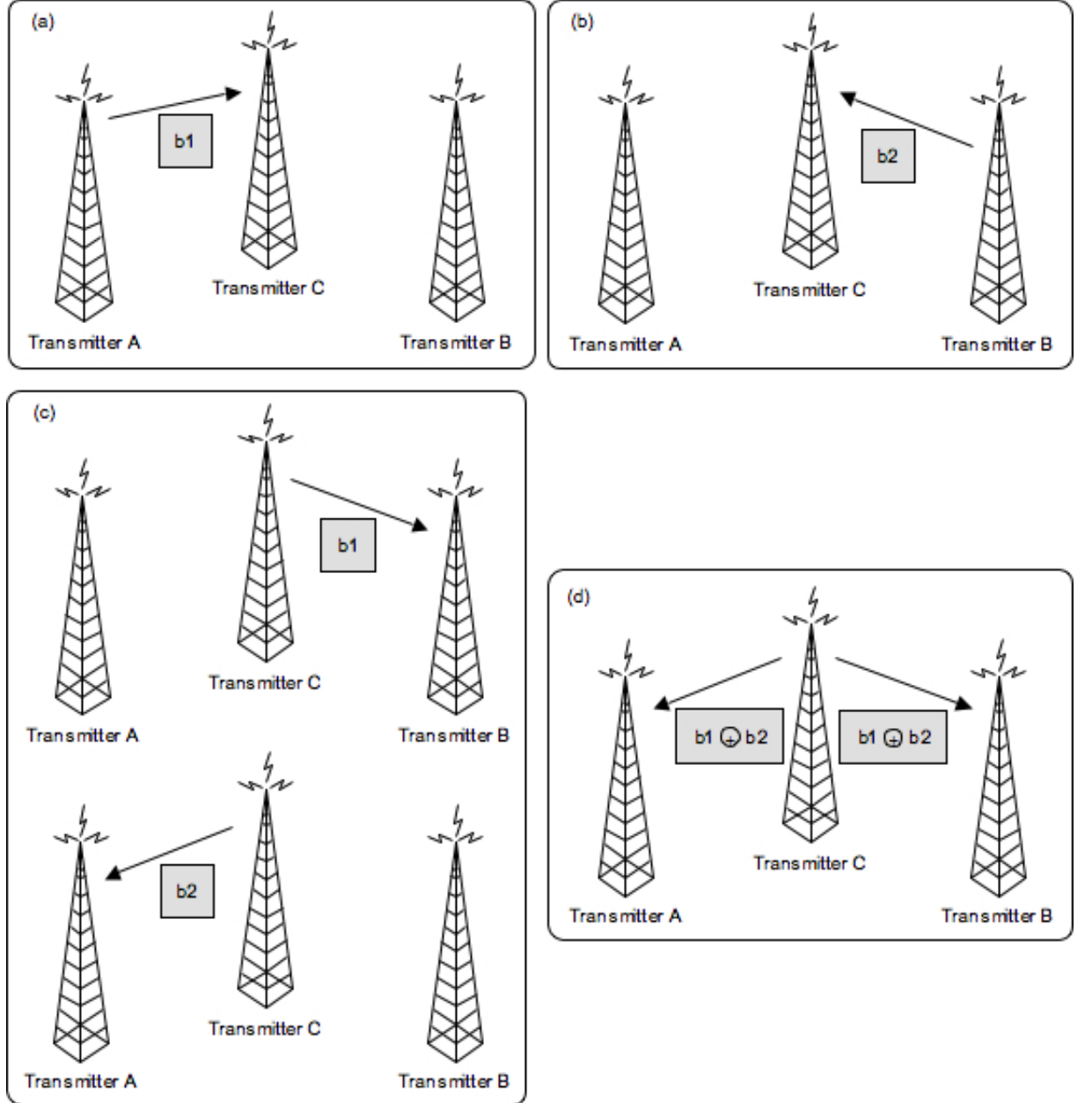


Figure 5.4: Network Coding. The figure shows the transmission of two data packets with the classical communication paradigm compared to the network coding technique. If two nodes are communicating with an intermediary node, they can benefit of this coding technique, thus optimizing the use of the channel and gaining higher throughput (in the example, nodes A and B want to communicate using node C as relay agent; while with the classical paradigm node C has to use the channel twice in order to send the flows $b1$ to B and $b2$ to A , with the network coding technique only transmission of flow $b1 + b2$).

5. DATA DISSEMINATION SURVEY

Assume that each packet consists of L bits. When the packets to be combined do not have the same size, the shorter ones are padded with trailing 0s. We can interpret s consecutive bits of a packet as a symbol over the field $\mathbb{F}_{2s}$, with each packet consisting of a vector of L/s symbols. With *linear* network coding, outgoing packets are linear combinations of the original packets, where addition and multiplication are performed over the field $\mathbb{F}_{2s}$. Clearly, one of the reasons for choosing a linear framework is that the algorithms for coding and decoding are simple and well understood.

It is important to consider that linear combination is not a concatenation: if we linearly combine packets of length L , the resulting encoded packet also has size L . An encoded packet generally carries information about several original packets, but in contrast to concatenation, just by itself it does not allow to recover any part of the original packets. One can think of linear *network coding as a form of information spreading*.

5.3.4 The coding and decoding process

The coding process can be easily explained as follows. Assume that a number of original packets $M^1, \dots, M^n$ are generated by one or several sources. In linear network coding, each packet in the network is associated with a sequence of coefficients $g_1, \dots, g_n$ in $\mathbb{F}_{2s}$ and is equal to:

$$X = \sum_{i=1}^n g_i M^i \quad (5.1)$$

The summation has to occur for every symbol position, i.e., $X_k = \sum_{i=1}^n g_i M_k^i$, where M_k^i and X_k is the k^{th} symbol of M^i and X respectively.

In the example of figure 5.4, the field is $\mathbb{F}_{2s} = \{0, 1\}$, a symbol is a bit, and the linear combination sent by S after receiving $M^1 = b_1$ and $M^2 = b_2$ is $M^1 + M^2$ (where the sign "+" is to be intended as the bitwise *xor*, the addition in $\mathbb{F}_{2s}$).

For sake of simplicity, we can assume that a packet contains both the coefficients $g = (g_1, \dots, g_n)$, called *encoding vector*, and the encoded data $X = \sum_{i=1}^n g_i M^i$, called *information vector* (111). The encoding vector is used by recipients to decode the data, as explained later. For example, the encoding vector $e_i = (0, \dots, 0, 1, 0, \dots, 0)$, where the 1 is at the i th position, means that the information vector is equal to M^i .

Encoding can be performed recursively, namely, with already encoded packets. Consider a node that has received and stored a set $(g^1, X^1), \dots, (g^m, X^m)$ of encoded packets, where g^j (and respectively X^j) is the encoding (respectively, *the information*) vector

of the j th packet. This node may generate a new encoded packet (g', X') by picking a set of coefficients $h_1, \dots, h_m$ and computing the linear combination $X' = \sum_{j=1}^m h_j X^j$. The corresponding encoding vector g' is not simply equal to h , since the coefficients are with respect to the original packets $M^1, \dots, M^n$; in contrast, straightforward algebra shows that it is given by $g'_i = \sum_{j=1}^m h_j g_i^j$. This operation may be repeated at several nodes in the network.

As it concerns the decoding phase, let us assume a node has received the set

$$(g^1, \dots, X^1), \dots, (g^m, \dots, X^m) \quad (5.2)$$

In order to retrieve the original packets, it needs to solve the following linear system with m equations and n unknowns (M^i):

$$\{X^j = \sum_{i=1}^n g_i^j M^i\} \quad (5.3)$$

In order to recover all data, we need $m \geq n$, which means that the number of received packets needs to be at least as large as the number of the original packets. Conversely, the condition $m \geq n$ is not sufficient, as some of the combinations might be linearly dependent.

At this point, it should also be clear that one of the main problem in network coding is how to select linear combinations that each node of the network performs. A simple algorithm is to have each node in the network which selects uniformly and randomly the coefficients over the said field $\mathbb{F}_{2^s}$, in a completely independent and decentralized fashion (36).

With random network coding there is a certain probability of selecting linearly dependent combinations which is mainly related to the field size 2^s (36). In (107), simulation results indicate that even for small field size (i.e., $s = 8$) the probability becomes negligible.

Alternatively, it is possible to use deterministic algorithms to design network codes: the polynomial-time algorithm for multicasting in (85) sequentially examines each node of the network and decides what linear combinations each node performs. Since each node uses fixed linear coefficients, packets only need to carry the information vector.

5. DATA DISSEMINATION SURVEY

5.3.5 Benefits of Network Coding

One of the main advantages that sparked the interest in the network coding technique is that it can increase the capacity of a network for multicast flows. In (9) we can find an important theorem which states:

theorem. *Assume that the source rates are such that, without network coding, the network can support each receiver in isolation (i.e. each receiver can decode all sources when it is the only receiver in the network). With an appropriate choice of linear coding coefficients, the network can support all receivers simultaneously.*

Let us consider a network that can be represented as a directed graph, where the vertices correspond to channels and nodes to terminals, respectively, with M sources and N receivers (all interested in receiving data from sources): the said theorem states that when the N receivers share the network resources, each of them can receive the maximum rate, even if it were using all the network resources by itself.

Network coding may offer throughput benefits not only for multicast flows, but also for other traffic patterns such as unicast. Anyway, the throughput gains of network coding for the former can be very significant (85).

An interesting point is that network coding allows to achieve the optimal throughput when multicasting using polynomial time algorithms. In contrast, achieving the optimal throughput with routing is NP-complete: this is *the problem of packing Steiner trees in CS theory*. Thus, even when the expected throughput benefits of network coding are not large, we expect to be able to achieve them using simpler algorithms.

Another very important benefit of network coding technique is given in terms of *robustness* and *adaptability*.

Intuitively, we can think that network coding, similarly to traditional coding, takes information packets and produces encoded packets, where each encoded packet is equally important. Provided we receive a sufficient number of encoded packets, no matter which, we are able to decode. The new twist that network coding brings, is that the linear combining is performed opportunistically over the network, not only at the source node, and thus it is well suited for the (typical) cases where nodes only have incomplete information about the global network state. Consider again Figure 5.4 and assume that A and B may go into sleep mode (or may move out of range) at random and without notifying the base station C . If the base station C broadcasts b_1 (or b_2),

the transmission might be completely wasted, since the intended destination might not be able to receive. However, if the base station broadcasts $b1 \text{ xor } b2$, or more generally, random linear combinations of the information packets, the transmission will bring new information to all active nodes.

Generally speaking, the network coding technique applied to wireless networks can improve throughput when two nodes communicate via a common base-station (cfr. 5.4). This setting can be extended to the case of multi-hop routing in wireless network, where the traffic between the two end nodes is bidirectional and both nodes have similar number of packets to exchange (106). In (106), the authors discuss a distributed implementation that works when transmissions are not synchronized and the wireless channel is lossy and has random delay. Overhearing a packet of a neighbor that is coded over information previously forwarded to the neighbor serves as a passive acknowledgment. This allows to make better use of transmission opportunities at routers that only have new packets buffered for a single direction. In this case, one of these new packets is combined with an old packet for the opposite direction, for which no passive acknowledgment has been received.

Another important gain in terms of network performance in wireless network given by network coding is the efficient implementation of many-to-many broadcast paradigm (used, for example, in route discovery), which may lead to significant energy savings(31).

In (32), authors simulate data dissemination and collection for sensor networks using a forwarding algorithm that sends out a packet with a certain probability, whenever a packet is received for the first time (flooding) or an innovative packet is received (network coding). When no data is disseminated and each node holds only its own packets, network coding and flooding obviously have the same performance. In case data is disseminated such that all nodes can decode all packets and a data collection phase is unnecessary, network coding offers a constant benefit in terms of required number of transmissions (31). Between these two extreme cases, network coding provides higher performance gains in terms of number of polled nodes.

For the scenario used in (32), authors vary the number of nodes in the network, as well as the network size, to have on average 12 neighbors per node. The total number of transmissions per node is limited and they count, how many nodes have to be polled by the data collector (sink) after the data dissemination phase, to obtain all data. Even

5. DATA DISSEMINATION SURVEY

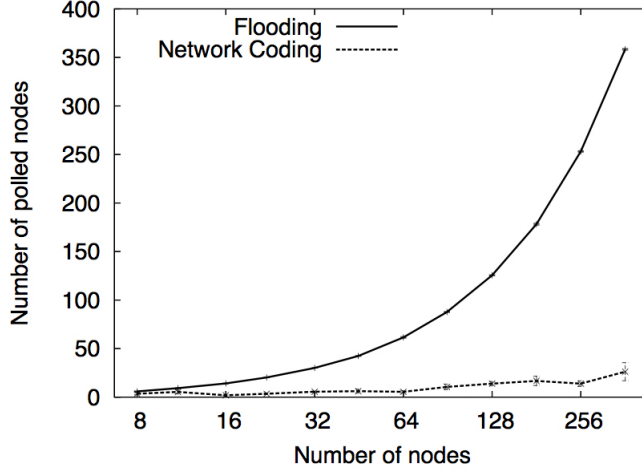


Figure 5.5: Network Coding benefits in data dissemination. The figure shows the number of nodes that have to be polled by the data collector to obtain all packets.

if nodes are allowed to transmit only once, network coding benefits a lot in very small scenarios but has little impact in larger networks, as shown in Fig.5.6.

As long as the allowed number of transmissions is significantly below the number of nodes in the network, flooding needs to poll almost all nodes. Here, flooding with 256 transmissions per node achieves roughly the same performance as network coding with 4 transmissions per node. For more than 16 transmissions per node, network coding delivers all packets to the sink, such that no polling is necessary. For the next set of simulations, they set the number of packets each node is allowed to transmit to $0.25\sqrt{n}$ (n is the number of nodes). In that setting, flooding takes almost no advantage of the proactive data dissemination (graph of Fig.5.5). The number of polled nodes is reduced by slightly more than the number of packets that each node transmits. In contrast, with network coding the sink node needs to poll only a few other nodes to be able to decode all original packets.

Other scenarios that benefit from the said technique (but are beyond the scope of this document) are: *data gathering in sensor networks* (23), *network tomography* (30) and *network security* (37)(27)(34).

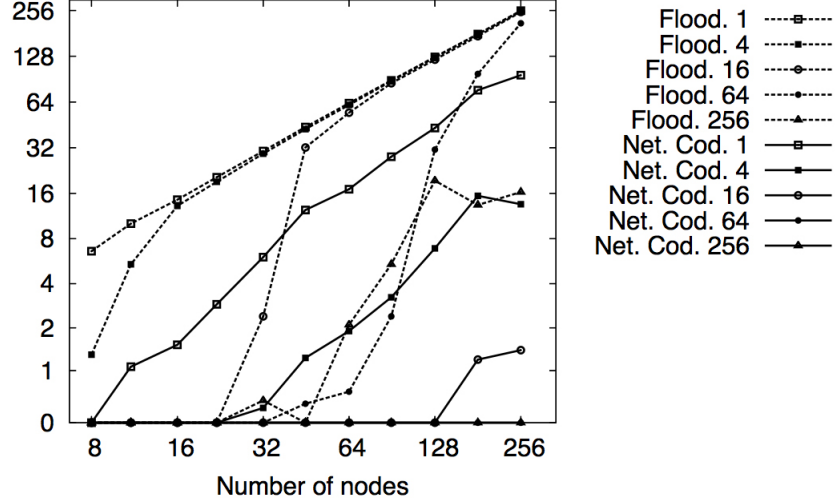


Figure 5.6: Network Coding benefits in data dissemination. Comparison between flooding and network coding: number of polled nodes for different numbers of transmission per node.

5.4 Example of Content distribution applications for VANETs

5.4.1 SPAWN (CarTorrent)

SPAWN (68) is a cooperative strategy for content delivery and sharing in VANETs, designed to improve client perceived performance of vehicular network as a whole. The key contributions of this protocol are:

- *a gossip mechanism* to propagate content availability information;
- *a proximity driven content selection strategy*, which takes into account the fact that TCP throughput degrades over multi-hop wireless connection;
- *leveraging the broadcast nature of wireless networks* to reduce redundant message transmission.

The basic structure of SPAWN is the same for every swarming protocol, that means peers downloading a file form a mesh and exchange pieces of the file amongst themselves. In order to perform an operation in SPAWN, there are three main phases to be taken into account:

5. DATA DISSEMINATION SURVEY

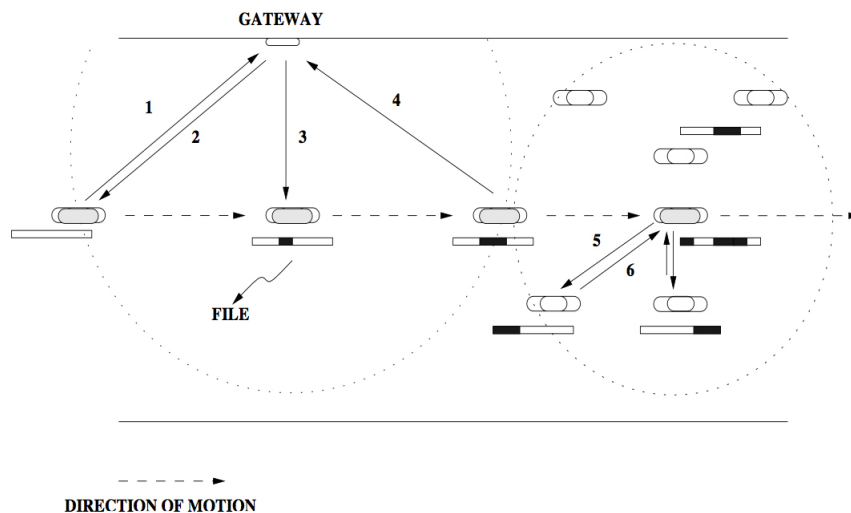


Figure 5.7: Evolution of a file in a node using SPAWN protocol. (1) A car arrives in the range of a gateway, (2) starts a download, (3) download a piece of a file, (4) after getting out of range, (5) starts to gossip with its neighbors about content availability and (6) exchanges pieces of the file, thereby getting a larger portion of the file as opposed to waiting for the next gateway to resume the download.

1. Peer Discovery

A new node enters the vehicular network and requests the gateway for the particular file. If the gateway has the file in its cache, it starts uploading a piece to the node. Decision policies with respect to piece choice are:

- *interested:*
nodes are interested in obtaining that particular file;
- *uninterested:*
the nodes are downloading another file, thus they are not interested in this one;
- *relays:*
these nodes do not understand the SPAWN protocol and, then, they drop the received packets. However, as they run some routing protocol (i.e. to support the general vehicular network or other safety applications running on top), they will relay IP packets which are unicast through them, even those associated with SPAWN file transfer.

5.4 Example of Content distribution applications for VANETs

At this point, the node starts downloading pieces from the gateway while it is in range. The mechanism for peer discovery is decentralized by using the broadcast medium of the wireless channel to gossip information about the content availability at neighbors. In a mobile environment, gossiping provides a way to incorporate location awareness into the peer discovery scheme. Since TCP over multiple-hops suffers quite dramatically in the ad-hoc wireless scenario (33), a node is better off using unicast TCP connections with peers who are just 4 or 5 hops away. Hence in SPAWN, the centralized approach and the gossiping mechanism can be used in conjunction to construct the mesh of peers and update connectivity information. The gossiping mechanism is introduced to increase the robustness of peer list maintenance and discovery, in the presence of high degree of node churn, as well as the intermittent presence of gateways.

2. *Peer and Content Selection*

SPAWN protocol does some intelligent Peer Selection based on the distance of the peer possessing a certain piece it intends to download. This information is gathered from the gossip messages but also from GPS enabled traffic safety messages that are likely to become standard applications running on vehicles in the future.

As it concerns the content selection, a proximity-driven piece selection strategy, which uses several distinct strategies to choose the pieces to download, is used.

3. *Content Discovery and Selection*

When two peers become neighbors, they exchange bitfields that describe which pieces they possess. When a peer fully receives a piece, it immediately notifies all of its neighbors. Thus, every peer always knows exactly which pieces its neighbors possesses. SPAWN uses UDP for delivery of gossip messages and TCP for content delivery.

SPAWN proves that the adoption of gossiping protocol in a VANET scenario offers several advantages. However, although analytical models have been developed, the simulation results are rather unrealistic because:

- the mobility model adopted in the simulation is too simple: cars maintain the same direction, almost the same speed and do not consider the cars around them;

5. DATA DISSEMINATION SURVEY

- simulation parameters are themselves unrealistic: 5MB file size, 64KB data fragments and simplified version of the 802.11 DCF protocol.

5.4.2 CodeTorrent

CodeTorrent is a solution born to investigate the problem of data dissemination in vehicular networks. In more detail, this protocol tries to apply the P2P paradigm to VANETs by the use of network coding technique and mobility assisted data propagation as the main ingredient.

Clearly, if a node has to send a file F (seed node), creates and then broadcast to its 1-hop neighbors the description of the file. Further to this, F is splitted into n pieces and then encoded, according to the linear network coding technique. At destination, in order to recover n file pieces $p_1, p_2, \dots, p_n$, a node must collect more than n coded frames carrying encoding vectors that are linearly independent of each other. Not only the seed node of a file, also every node which possesses any coded frame of the file and willing to share them periodically broadcasts (at a very low rate) to its 1-hop neighbors the description of the file. If a node has multiple files to share, multiple descriptions are packed into the least number of packets that can carry all of them and then transmitted. A request of coded frames may be accompanied by the null space vector which is a vector in the nullspace spanned by all encoding vectors of the frames stored in the local memory of the requesting node. On reception of such a request, a node transmits a coded frame only if there is in its local memory a frame with the encoding vector that is not orthogonal to the nullspace vector received with the request. Every node promiscuously listens to packets, i.e., a node receives a specific packet even the node is not the designated receiver, so that it can use them if possible. A node always overhears the packets carrying coded frames and treats the overhead ones as the coded frames transmitted specifically to the node. If an overheard coded frame is linearly independent of the coded frames in local memory, then a node stores it.

In CodeTorrent the average download delay decreases as mobility increases and, since this protocol is based on single hop data pulling and overhearing, mobility plays an important role such that *data dissemination latency could be reduced with increased mobility*. The main issues related to CodeTorrent (like almost every data dissemination protocol for VANETs) are the number of vehicles involved in the communications and

5.4 Example of Content distribution applications for VANETs

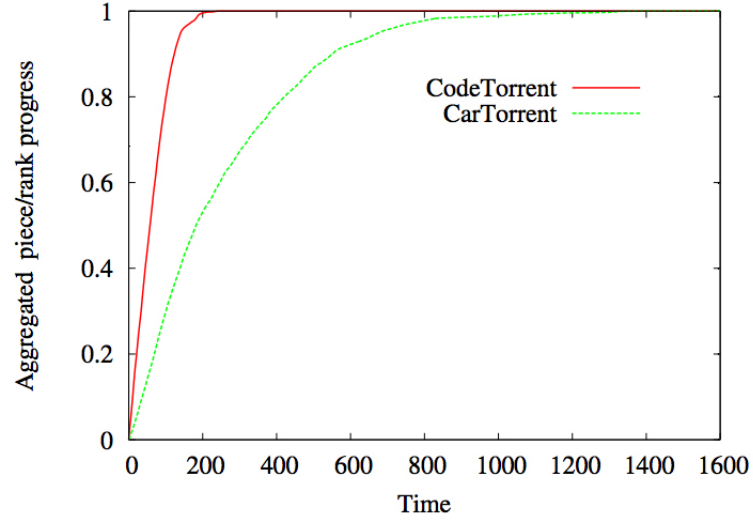


Figure 5.8: CodeTorrent vs. CarTorrent. The figure shows the aggregated downloading progress (200 nodes moving with the maximum speed of 20 m/s). The number of interested nodes is 80 (which means that the *popularity index* is 40%).

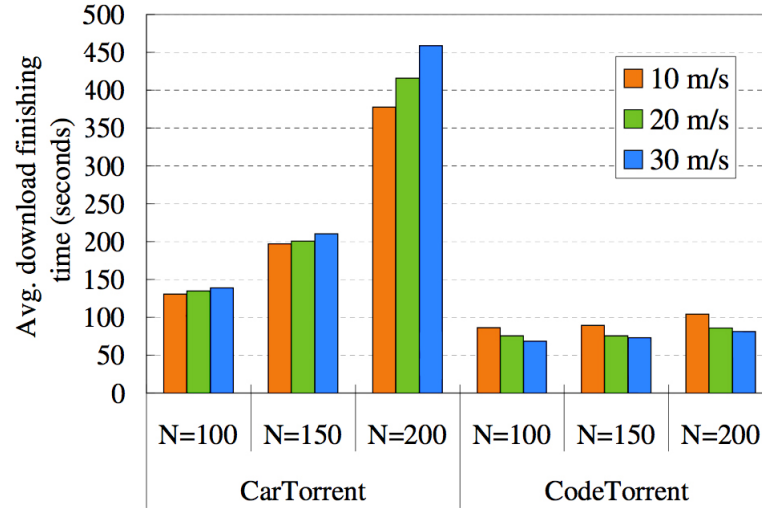


Figure 5.9: CodeTorrent vs. CarTorrent. Impact of mobility on average download delay.

5. DATA DISSEMINATION SURVEY

the velocity of mobile nodes. In fact, the number is important for information dissemination but congestion is inevitable as the number of nodes participating gossiping increases. As it concerns the mobility, if it can reduce the dissemination latency on one hand, on the other it can also affect the overall performance because the contact period between two nodes becomes too short to exchange a data fragment.

5.4 Example of Content distribution applications for VANETs

#	Data Element	# of bits	Signal Violation	Curve Warning	Emerg. Brake	Pre-crash Sensing	Collision Warning	Turn Assist.	Lane Warning	Stop-Sign Assist.	# of uses
1	acceleration	12	•	•	•	•	•	•	•	•	8
2	acceleration precision	3	•	•	•		•	•		•	6
3	airbag count	7					•				1
4	ambient air temp.	8									0
5	anti-lock brake state	2	•		•		•				3
6	brake applied pressure	4	•	•	•		•	•		•	6
7	brake applied status	4	•	•	•		•	•		•	6
8	brake boost applied	1		•	•	•					3
9	driving wheel angle	8		•	•			•			3
10	DSRC message ID	12	•	•	•	•	•	•	•	•	8
11	elevation confidence	4		•							1
12	elevation	20		•		•					2
13	exterior lights	3			•						1
14	heading	16	•	•	•	•	•	•	•	•	8
15	heading precision	3	•	•	•	•	•	•	•	•	8
16	headlights	2						•			1
17	lateral acceleration	12	•		•	•		•	•	•	8
18	latitude of center of vehicle	32	•	•	•	•	•	•	•	•	8
19	longitude of center of vehicle	32	•	•	•	•	•	•	•	•	8
20	obstacle direction	16		•		•	•				3
21	obstacle distance	10	•	•	•	•	•	•	•	•	8
22	longitudinal acceleration	12	•	•	•	•	•	•	•	•	8
23	positioning precision	4	•	•	•	•	•	•	•	•	8
24	rain sensor	3		•				•			3
25	siren in use	2	•								1
26	speed	13	•	•		•	•	•	•	•	7
27	speed precision	3	•	•		•	•	•	•	•	7
28	stability control status	3		•	•	•	•	•			5
29	steering wheel angle	16	•	•			•		•		4
30	steering wheel angle precision	2	•	•			•		•		4
31	steering wheel rate of change	8			•	•	•	•	•		8
32	sun sensor	10									0
33	system health	4				•					1
34	throttle position	8	•	•			•	•		•	5
35	throttle precision	3	•	•			•			•	4
36	time precision	4	•	•		•	•		•	•	6
37	temporary ID	48					•		•		2
38	traction control state	2		•					•		2
39	turn signal/hazard signal	2	•				•	•	•	•	5
41	UTC time	40	•			•	•	•	•	•	5
42	vehicle length	14				•	•	•	•	•	5
43	vehicle width	10				•	•		•		3
44	vehicle height	8	•						•		2
45	vehicle mass	8		•	•	•	•				4
46	vehicle type	7		•		•		•			3
47	vertical acceleration	8									0
48	vertical acceleration threshold	4									0
49	wiper rate	8						•			1
50	wiper status	3						•			1
51	yaw rate	16		•							1
52	yaw rate precision	3		•							1

Table 5.2: A subset of the SAE common message set (more than 70 data elements) and their usage in vanet safety applications.

5. DATA DISSEMINATION SURVEY

Chapter 6

Data dissemination with rateless codes

6.1 Before you start: digital fountain codes

Digital Fountain (DF) codes are random sparse-graph codes developed for erasure channels¹ and they can be compared to a running tap: when a cup is filled it is not important which droplets land in the cup, but only that enough water is required to fill the cup. The above metaphor also highlights another important property of digital fountain codes namely its ability to generate an infinite amount of encoded packets from the original source.

Actually, a packet layer communication can be modeled as an erasure channel, so that these codes can be used as FEC² codes on data packets. Therefore, Fountain codes can be used to enhance performance of the system in lossy channels, where classical approaches typically work poorly. In fact, in these channels, TCP protocol is not efficient because of the high amount of retransmissions required to receive the source. In

¹An erasure channel is a channel in which each codeword symbol is lost with a fixed constant probability p in transit independent of all the other symbols.

²In telecommunication and information theory, Forward Error Correction is a system of error control for data transmission, whereby the sender adds redundant data to its messages, also known as an error correction code. This allows the receiver to detect and correct errors (within some bound) without the need to ask the sender for additional data. The advantage of forward error correction is that a back-channel is not required, or that retransmission of data can often be avoided, at the cost of higher bandwidth requirements on average. FEC is therefore applied in situations where retransmissions are relatively costly or impossible. In particular, FEC information is usually added to most mass storage devices to protect against damage to the stored data.

6. DATA DISSEMINATION WITH RATELESS CODES

similar system conditions, UDP protocol might also perform poorly because it does not check the source integrity, that will be then granted by the upper layers. A solution to this problem can be a previous rateless encoding of the source data and a followed by a transmission using UDP. This would eliminate the need for retransmission since it is only critical that the decoder receives enough symbols. By using these codes the source data can be recovered from any subset of encoded packets, given that enough packets are received. A DF encoder generates can be thought of as a fountain that produces an endless supply of water drops. The water drops are encoded symbols (ESs). Similarly, a DF decoder can be thought of as a bucket that collects water drops until it reaches capacity. Given that the number of symbols that can be generated from the source data is potentially infinite and that every symbol can be generated on the fly, it is not possible to determine the rate in advance. Thus, these codes are also called rateless.

The generation of an ES is based on GF(2). In particular, each EC is obtained by XOR operations (ex-oring) on a certain number of source symbols chosen uniformly at random. The number of source symbols that are chosen is given by a outer degree distribution that relies on the particular code. Examples of this distribution can be found in (58; 66). The probability of generating two or more ESs with the same information is negligible. Consequently, each ES has equal importance and, on average, it carries the same amount of information as the others. The number of different symbols to be collected from the receiver in order to recover the source info is $(1 + \epsilon) \cdot k$, where k is the number of source symbols and ϵ is the coding inefficiency (usually, we have that $\epsilon \ll 1$). However, the impact of this inefficiency on the overall performance of the system is the lower the more lossy is the channel. In fact, It can be demonstrated that for $k \rightarrow \infty$ these codes have $\epsilon \rightarrow 0$, so they are potentially optimal(58). This means that, apart from the decoding inefficiency, it is possible to recover the source information as soon as enough ESs are received. Moreover, these codes are universal because the symbol length for the codes can be arbitrary. In fact, a symbol can extend from one-bit to general l bits, without affecting the coding and decoding efficiency.

6.1.1 LT codes

Luby Transform (LT) codes were the first rateless code used to approximate a fountain. They are a class of asymptotically optimal rateless erasure correcting codes introduced by Michael Luby in 2002 (58). Similarly to LDPC (low density parity check) codes,

LT codes rely on a sparse bipartite graph representation to trade reception overhead for encoding and decoding speed. These codes transmit random linear combinations of message symbols, where the number of message symbols contributing to each transmitted symbol is chosen at random with a given probability distribution.

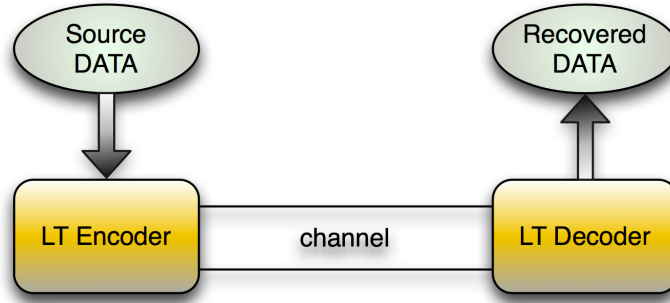


Figure 6.1: LT code. Functioning scheme of LT codes.

LT codes are rateless, which means that the number of encoded symbols that can be generated from the data is potentially limitless. In other words, their design does not depend on the estimate of erasure probability of the channel, so they can simultaneously serve a heterogeneous population of receivers efficiently. Furthermore, distinct encoding symbols can be generated on the fly by the server as needed. As a consequence, in order to let the decoder know which message nodes are the neighbors of a particular encoding symbol, Luby suggests that one solution is to explicitly include this information as an additional overhead in the packet. Another possibility is to replicate the pseudorandom process at the receiver by supplying it with the suitable seed and/or keys (i.e, as applied in CORP protocol). The k original data blocks are obtained by partitioning the original data into k uniform segments of l bits each. An encoded symbol is generated in the following manner:

1. *choose the degree d , that is the number of source symbols that form the encoded symbol. The selection of the degree is random, given a degree distribution ρ ;*

6. DATA DISSEMINATION WITH RATELESS CODES

2. *uniformly choose the d symbols that will contribute to the coded info. This symbols will be named neighbors;*
3. *the coded symbol info is obtained from a bitwise exclusive-or of the d neighbors.*

Description	Expression
message symbols	n
probability to reconstruct message symbols	$1 - \delta$
symbols to receive	$n + O(\sqrt{n} \cdot \log^2(n/\delta))$
time for decoding each symbol	$\propto O(\log(n/\delta))$

Table 6.1: Main characteristics of an LT code.

The effect of the said procedure is exactly the same as multiplying a message with a dynamic random G matrix.

The choice of the degree i has to follow a particular probability distribution ρ . In particular, the probability of selecting a degree i is ρ_i , so that the distribution can be dened as $\rho = (\rho_1, \rho_2, \rho_3, \dots, \rho_k)$, where $\sum_{i=1}^k \rho_i = 1$.

The design of a good degree distribution is very important in order to achieve good performance. Luby proposed to use the *Robust Soliton Distribution*, like presented in [(58)] that depends on the block size k and is characterized by the two parameters δ and c . By using this distribution, the creation of each encoded symbol requires $O(\log(k/\delta))$ logical ex-or operations per each generated symbol.

The Robust Soliton distribution ensures that the ripple size, i.e. the number of degree-one encoded symbols, is large enough to not extinguish, but it also assures that the ripple is not too large in order to avoid redundant packets. In this way, in order to decode the original data with a $1 - \delta$ chance of success, any $k + O(\sqrt{k} \cdot \log^2(k/\delta))$ encoded symbols are sufficient.

In order to decode a fountain coded message, we have to go through the following steps:

1. find a parity bit p_n that is connected to only one message bit m_k (if there is no such parity bit, the decoding algorithm stops and fails to recover all message bits);

- (a) Set $m_k = p_n$;
- (b) Add m_k to all parity bits $p_{n'}$ that are connected to m_k :
 $p_{n'} := p_{n'} + m_k$ for all n' such that $G_{n'k} = 1$;
- (c) remove all the edge connected to m_k ;

2. Repeat step 1 until all m_k has been recovered.

In order to better understand the said procedure, please consider the example of Fig. 6.2, which represents the decoding process of a very simple LT code on erasure channel. In the image, there are three message bits ($b1, b2, b3$) and four parity bits ($p1, p2, p3, p4$), which at the very beginning of the decoding phase are set to the vector [1011]. According to the algorithm, during the rst iteration, the only parity bit that is

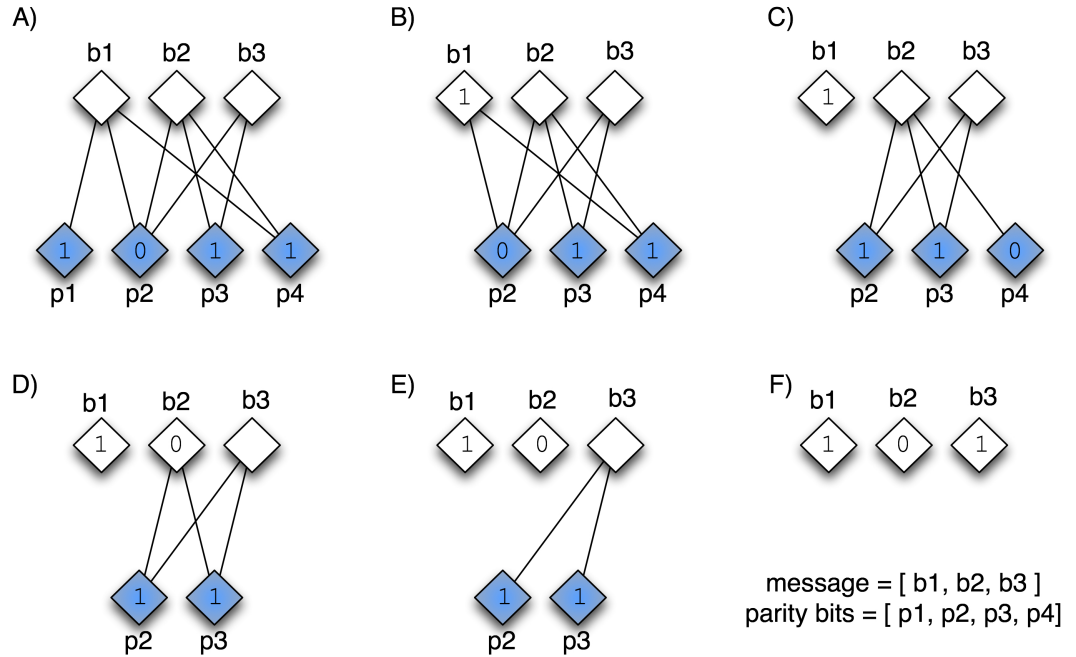


Figure 6.2: LT example. Decoding of a simple LT code on an erasure channel.

connected to one message bit is $p1$ (see Figure 6.2-A), whose value is then copied to $b1$ (see Figure 6.2-B) and then the relevant parity bit removed from the graph. The new value of $b1$ is added (xor-operation) to $p2$ and $p4$ and thus becoming disconnected from the graph. At the start of the second iteration (Fig. 6.2-C) the value of $p4$ is copied

6. DATA DISSEMINATION WITH RATELESS CODES

into message bit $b2$ as it is the only bit connected to it (Figure 6.2-D), and then add this value to $p2$ and $p3$ (Figure 6.2-E). Finally, it is clear that the parity bits connected to $b3$ are equal as expected and can be used to restore $b3$ (see Figure 6.2-F).

An important property of DF codes is that the probability of generating two identical ESs is negligible. In particular, for LT codes we can compute this probability as follows. Let us suppose that N ESs are generated and that the ratio of symbols with degree i is equal to ρ_i . Then, we can define the number of ESs of degree i equal to $h_i = N \cdot \rho_i$. Obviously, we have that $\sum_{i=1}^k h_i = N$.

Since the possible number of encoded symbols of degree i are $\binom{k}{i}$, thus *the probability that two i -degree ESs are identical* is:

$$\frac{\binom{k}{i}!}{\binom{k}{i}^{h_i} \cdot \left(\binom{k}{i} - h_i\right)!} = \frac{\prod_{j=0}^{h_i-1} \left(\binom{k}{i} - j\right)}{\binom{k}{i}^{h_i}} \quad (6.1)$$

From (6.2) we can easily evaluate *the probability that non of the generated ESs is repeated*:

$$\prod_{i=0}^k \frac{\prod_{j=0}^{h_i-1} \left(\binom{k}{i} - j\right)}{\binom{k}{i}^{h_i}} \quad (6.2)$$

6.1.2 Tornado codes

Tornado codes are erasure block codes based on irregular spare graphs. Given an erasure channel with loss probability p , they can correct up to $p \cdot (1 - \epsilon)$ errors. They can be encoded and decoded in time proportional to $n \cdot \log(1/\epsilon)$. Thus, Tornado codes has been primarily designed to speed up erasure codes over the internet. These codes can be designed over arbitrary alphabet size.

Clearly, the main benefit of Tornado codes, is the linear time decoding, which is the result of a very simple decoding algorithm. At each step, a right node is selected whose all but one neighbors are known. The missing neighbor is then computed by performing a XOR operation beetwen the check bit and the known inputs bits. If at any step, no such node is found, then an error occurs. The algorithm terminates successfully when all the input bits are recovered.

Obviously, the algorithm shows a linear time, since at each step we recover one lost

input bit and it is also not optimal, since it can fail even when it would have been possible to recover the input bits through say gaussian elimination.

6.1.3 Raptor codes

Raptor codes, presented by Shokrollahi in (89), extend the idea of LT codes one important step further. In fact, one of the features of LT codes is that they need an average degree $O(\log k)$ to code at least once every source symbol with high probability. The solution implemented by Raptor codes is to pre-code the message M by means of a fixed length erasure code, such as a Tornado code (17) or LDPC codes (83), leading to a new encoded message M' which is encoded with LT codes. Hence, Raptor Codes are LT codes with a preventive encoding as shown in fig.6.3.

The main advantage of the precoding stage is that, for correctly decoding, it is not

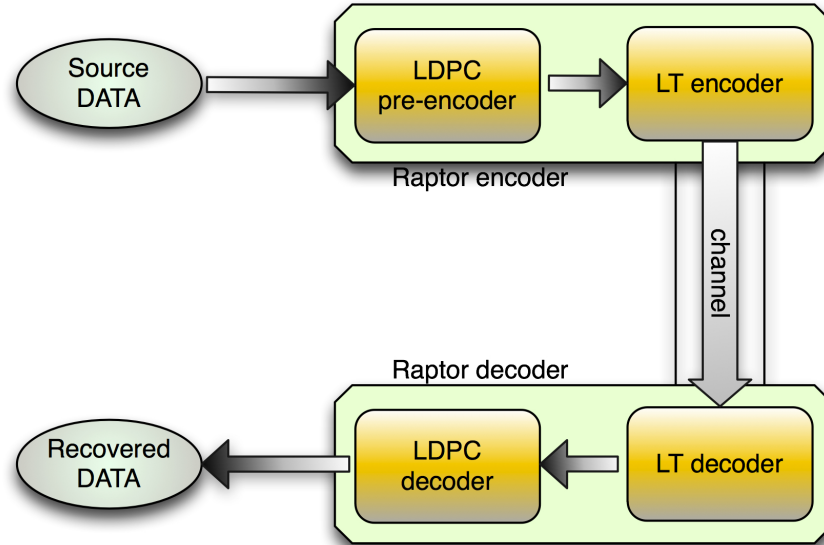


Figure 6.3: RAPTOR coding. Functioning scheme of RAPTOR codes.

necessary to decode all symbols of the message M' , but only a constant fraction of these. Thus, it is possible to use a simpler degree distribution that does not recover all the symbols but makes the decoding process faster. By using a precoding stage and a simpler degree distribution, the bound $O(\ln(k))$ on the average degree of LT codes no longer applies. Indeed, thanks to pre-coding with an appropriate design, for any

6. DATA DISSEMINATION WITH RATELESS CODES

constant $\epsilon > 0$ and sufficient large k , the original message M can be correctly decoded given that $(1 + \epsilon) \cdot k$ symbols have been received.

It can be proved that the value of ϵ tends to zero as k increases. This means that the property of optimality of digital fountains is verified also for these codes.

Shokrollahi proposed in (89) a new degree distribution that depends only on the overhead and has a maximum degree far lower than k . In particular, the average degree is no longer dependent on the logarithm of k , but it is constant, being $O(\ln(1/\epsilon))$. The constant degree leads to a total decoding time linear with k , i.e. $O(k \cdot \ln(1/\epsilon))$. In the following, we will refer to the new distribution as *Shokrollahi distribution* and to the new LT code as *weakened LT code (wLT)*. In fig.6.4 we can observe a comparison between Robust Soliton Distribution and Shokrollahi Distribution. As we can observe, the second peak is lower and the probability of symbols with degree one or two is much larger than using RSD.

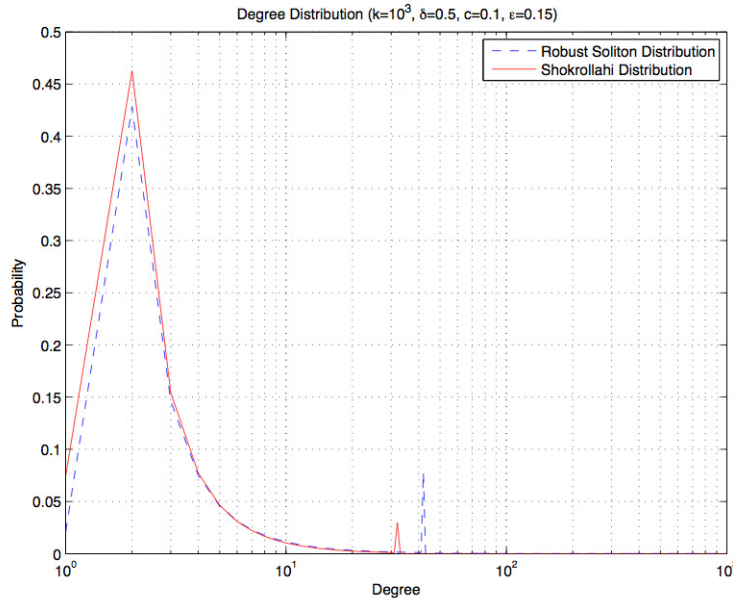


Figure 6.4: RS-SD comparison. Comparison between Robust Soliton and Shokrollahi Distributions.

The decoding algorithm is composed of two steps. The inner LT decoder returns a hard bit-reliability vector. This latter is processed by the outer LDPC decoder, based on belief propagation algorithm. The main drawback of Raptor codes is that the total

6.1 Before you start: digital fountain codes

overhead is lower bounded by the overhead of the pre-code. On the contrary, LT codes have no overhead asymptotically.

Code	Negatives	Positives
<i>RS</i>	– Limited distinct decoding symbols – Quadratic decoding algorithm	– Can decode with k encoded packets
<i>Tornado</i>	– Need more than k encoded packets to decode – Limited distinct decoding symbols	– Decoding time proportional n
<i>LT</i>	– Need more than k encoded packets to decode	– Decoding time proportional to $k \cdot \log(n)$ – Unlimited distinct decoded symbols
<i>Raptor</i>	– Need more than k encoded packets to decode	– Decoding time proportional to k (due to pre-coding)

Table 6.2: Comparison of different coding schemes used to approximate a digital fountain code.

Moreover, similarly to LTcodes, Raptor Codes are not systematic. This means that the input symbols are not necessarily reproduced among the output symbols. The straight forward idea of transmitting the input symbols as a preamble of the symbols can be easily shown to be awed, since this does not guarantee a high probability of decodability from any subset of received output symbols (89). Shokrollahi introduced a new systematic version of Raptor Codes in (89) that has then further developed in (59).

In streaming applications, the effects of even small levels of packet loss are indeed apparent to the end user. In ahead-to-head comparison of Raptor and Reed-Solomon (RS) erasure codes, used to protect streaming applications, the following conclusions can be drawn:

- Raptor codes are more efficient than comparable RS code in protecting against packet loss.
- Raptor codes require significantly less processing power than Reed-Solomon erasure codes for encoding and decoding; the required processing power for Raptor

6. DATA DISSEMINATION WITH RATELESS CODES

codes increases linearly with the level of provided protection, whereas, for Reed-Solomon erasure codes, it increases quadratically.

- Raptor codes allow a given application to be optimally addressed in terms of the degree of packet loss protection, bandwidth expansion and processing demands.

6.2 An application of fountain codes: CORP

6.2.1 Introduction

This chapter presents CORP (19) (Cooperative Rateless Protocol), a new rateless protocol for data dissemination designed for vehicular networks which exploits the reliability of the rateless coding approach while performing fast and efficient dissemination through cooperating nodes within the network. The application that we consider for CORP is e-commerce and e-advertising. When a car approaches a pre-denied area, information about local commercial offers is provided. Such information may range from common goods to sophisticated service offerings and may also include events and similar entities. Via vehicular communications, vehicle passengers can request more information and, if electronic commerce is supported, consumers can immediately offer and continue with the transaction. As an example, a car driver may inquire about the price of a particular good at various local businesses and selects the best price/value offering. Alternatively, a local business can advertise to car drivers in a certain area a new offer (e.g., restaurant menu, etc.). CORP protocol implements the information dissemination process between local businesses and vehicles.

Current data dissemination protocols for vehicular networks still lack the reliability and the efficiency required by most applications, ranging from safe navigation to content delivery. One of the main problems is the unreliable connectivity. In fact, existing dissemination methods (e.g., BitTorrent) use TCP for data transmission among nodes. Unfortunately, due to high vehicular mobility, TCP does not perform well and cannot consistently guarantee efficient communications and as a result, frequent packet loss affects both window size and transmission efficiency.

On the other side, the use of UDP for data transfer may not be suitable for dissemination in vehicular networks. In particular, the lack of acknowledgements from the receiver may trigger repeated transmissions regardless of the status of the transfer. This

problem can be solved by allowing the receiver to send a packet to the sender to stop the data transmission when it has received the information. As is well known, UDP cannot guarantee the reliability of the data transfer. In fact, in the event that even a single packet is lost, the information will not be received correctly. The classic solution involves using *erasure-correcting* codes. By using these codes, a subset of encoded packets will allow to recover the original content. Among all the erasure-correction techniques presented in literature, rateless codes are the most interesting ones. In fact, they provide low encoding and decoding complexity and are especially suitable in networks where the channel conditions are unpredictable and unknown. These are often referred to as *channel oblivious* transmissions, because the number of encoded packets that can be generated is potentially infinite.

The use of erasure-correcting codes for the dissemination of data not only improves the reliability of the transmission but also helps to address the problem known as *coupon problem* (56). This problem is typical of P2P networks where the amount of time needed to obtain the last missing piece of the data can be very long. By generating encoded information and disseminating it cross the network, it is possible to increase the information availability.

Thanks to their unique characteristics, rateless codes have been widely used for data dissemination over the Internet. Yet, another problem that P2P systems face is how to efficiently exchange data among peers. At the beginning of the communication, peers have to exchange information about the packets they want to receive in order to efficiently use the available bandwidth by avoiding duplicates. In this work we will call this phase the content reconciliation phase, that is heavier, the larger the information is. *Speed* is a key factor in the design of a communication system in vehicular networks and is therefore critical that nodes exchange information about their contents as fast as possible. This means that the efficient techniques have to be developed in order to minimize the content reconciliation phase.

With CORP protocol, we aimed to address the following issues:

- reliable and fast data dissemination in vehicular networks;
- maximization of the content availability on the network;
- minimization of the content reconciliation phase cost.

6. DATA DISSEMINATION WITH RATELESS CODES

In addition to our primary goals, we also tried to design a protocol that does not demand high computing resources, thus extending our solution to portable devices. In the following sections, a preliminary analysis of the performance of CORP is presented. Basically, we focus on these three metrics:

1. delivery ratio after 15 minutes as a function of the nodes that collaborate in the dissemination;
2. delivery ratio as function of time;
3. average number and size of the sets of encoded packets that are exchanged among nodes.

For the analysis we adopted a first a random-waypoint mobility scenario (RWP) (22). Although this model is only an approximation of vehicles behavior, it is still useful to understand the asymptotical performance of the protocol. In order to provide more accurate results, additional tests have been performed with pseudo-real traces, generated with Vanetmobisim(35) on a real map imported from Open Street Map¹.

In section 6.2.7, CORP is compared to other two similar approaches. The first one, *Basic Rateless Protocol* (BRP) considers a rateless dissemination from a single access point to all the interested vehicles in its range, so that the nodes can recover the information if they have collected enough encoded packets. Since there is no cooperation among vehicles, the communication will be of the type infrastructure-to-vehicles (I2V). In the second one, mobile nodes can disseminate the source information as after having decoded it. Therefore, we will call this approach DDRP (i.e., *decode and disseminate rateless protocol*). Please, note that the latter is similar to the approach presented in (104).

6.2.2 Related work

The extant literature presents many proposed solutions for data dissemination over vehicular networks (introduced in (74) and (75)). The most interesting solution ex-

¹Open Street Map (OMS) is a free editable map of the whole world which allows you to view, edit and use geographical data in a collaborative way from anywhere on Earth. The website is: <http://www.openstreetmap.org>.

OMS hosting is kindly supported by the UCL VR Centre and bytemark. Other supporters of the project are listed in the wiki.<http://www.openstreetmap.org/>

exploits network coding in order to obtain high bandwidth efficiency. However, packet loss can deeply affect the performance of these methods. This problem has been recently addressed in (80), where errors and erasures can be tolerated. This method generates codes on a vector space and avoids the overhead of the encoded set description while simultaneously providing highly efficient transmissions. Unfortunately, this is not a desirable fit for our scenario as the the recoding phase of network coding approach is computationally more expensive than a smart selection of the received information to be forwarded to other nodes. Byers et al. in (43) study the content reconciliation problem in wired networks. In their study, two nodes compute the resemblance of their information in order to understand if the communication can be useful. The authors also present appropriate tools for an effective content reconciliation. Although the methods presented in their paper are very interesting, we argue that the performance in wireless vehicular networks would be poor. For example, to perform the content reconciliation methods, nodes would be required to periodically exchange packets to update the neighbors on its own representation of the information. Additional time would also be required to compute the information that is to be sent. In sum, the content reconciliation phase would require too much time.

Yet another solution, *Bullet*, is found in (50). This method can use either rateless or multiple description coding to efficiently disseminate data. It efficiently solves the problem of choosing the best nodes to talk to in an overlay mesh network. Moreover, it uses a simplified version of TCP to fulfill the requirements of multimedia transmission that is also TCP friendly. Performance analysis shows that *Bullet* is efficient on wired networks but again performs poorly in vehicular networks for reasons mentioned earlier. The work in (67) simplifies the content reconciliation phase by sequentially labeling the received packets and by exchanging the sequence number ranges in order to perform a simple content reconciliation. Unfortunately, this solution can work only in wired networks where packet loss is almost negligible. In fact, when loss happens, the number of subranges of packets to be transmitted grows, thus decreasing the efficiency.

Finally, a method for P2P streaming in wired networks, *rStream*, is presented in (104). *rStream* is based on the encoding of the source using rateless codes. According to this method, every node forwards the information after having decoded it. Furthermore, availability of information is assured if each node generates a set of symbols that is independent of the others in the network. This allows symbols to be received from

6. DATA DISSEMINATION WITH RATELESS CODES

several nodes at the same time without the need for content reconciliation. Unfortunately, a similar approach in vehicular networks would perform poorly because it is based on the implicit assumption that connections between source node and interested nodes have sufficient time to receive all the encoded information needed for recovery. However, this may not happen in vehicular networks because a node can be connected to APs or vehicles for only a limited time. Therefore, the transmission of the encoded information may not be quickly completed.

6.2.3 Proposed approach

The following Section presents the main ideas of the proposed protocol. The considered scenario is based on access points (APs) and mobile nodes (MNs). APs collect information from external sources and deliver it (or fractions of it) to interested MNs that will disseminate it to other vehicles. In other words, interested vehicles cooperate with each other to disseminate information. APs can be fixed or mobile nodes, or a mix of those. However, for simplicity, this paper considers only fixed APs. The dissemination approach can be divided in two different phases:

1. The first phase is the *dissemination between AP and MN*. This phase starts when a MN communicates its interest in a certain information that an AP is advertising.
2. The second phase pertains to the *dissemination among MNs*, in which nodes cooperate to disseminate the received data.

The dissemination of the information is performed by exploiting the characteristics of rateless codes. These codes have been presented in (58) and in (66) and implement the digital fountain idea.

From a block of K source information symbols, the rateless encoder can produce a potentially infinite amount of encoded symbols (ESs). The generation of the ESs is based on random ex-oring of source symbols that depends on a random sequence generator that is assumed to be known to both encoder and decoder (see section 5.1). The ESs generated from a random sequence (that is created from a particular *generation seed*) form a set of ESs that we call Λ . Given the random nature of the encoding process, the probability of generating two identical ESs is negligible for sufficiently large information blocks. This means that two different sets (i.e., created from two different

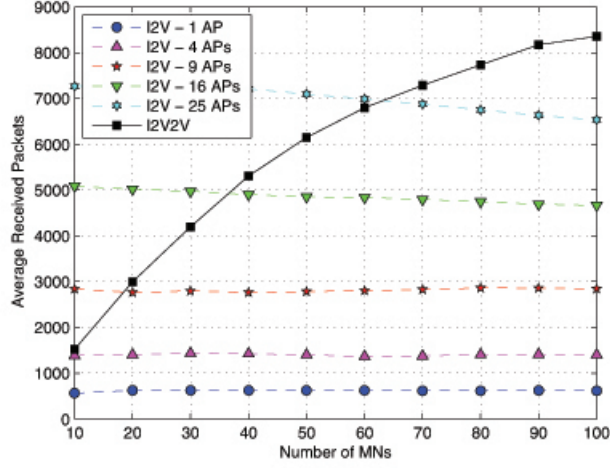


Figure 6.5: I2V vs. I2V2V The figure shows the benefit of an I2V2V approach (like in CORP) over the classical I2V, where data are disseminated only by APs.

seeds) contain different ESs with high probability, i.e., the sets are *disjoined*. As an example, we can consider a le of 512 KB composed of 8000 source symbols of 64 bytes. The block is encoded with different generation seeds, S_a, S_b, S_c , etc., thus generating two disjoined sets of ESs $\Lambda_a, \Lambda_b, \Lambda_c$, etc. Given that enough symbols have been received, the rateless decoder is able to recover the source block with high probability starting from any subset of distinct received ESs. In particular, the received ESs can belong to different sets of ESs without any loss of performance. The number of symbols N that have to be received is slightly greater than K . This means that the decoding process introduces an *inefficiency* $\epsilon = N/K - 1$. Usually, this inefficiency is very small and it can be demonstrated that it tends to be zero when the number of source symbols tends to infinite.

The proposed approach is based on the dissemination of sets of ESs, rather than on single packets. This means that for a given block, nodes try to exchange all the ESs belonging to a set that is not in common (e.g., Λ_a), before exchanging ESs from another set (e.g., Λ_b). This method is advantageous in that ESs are transferred in groups, thus reducing in this way the content reconciliation complexity. In fact, since sets are univocally identified by the generation seeds, the content reconciliation algorithm is reduced to the research of the seeds that are not in common between two communicating nodes. For example, let us suppose that a node has received the ESs belonging to the

6. DATA DISSEMINATION WITH RATELESS CODES

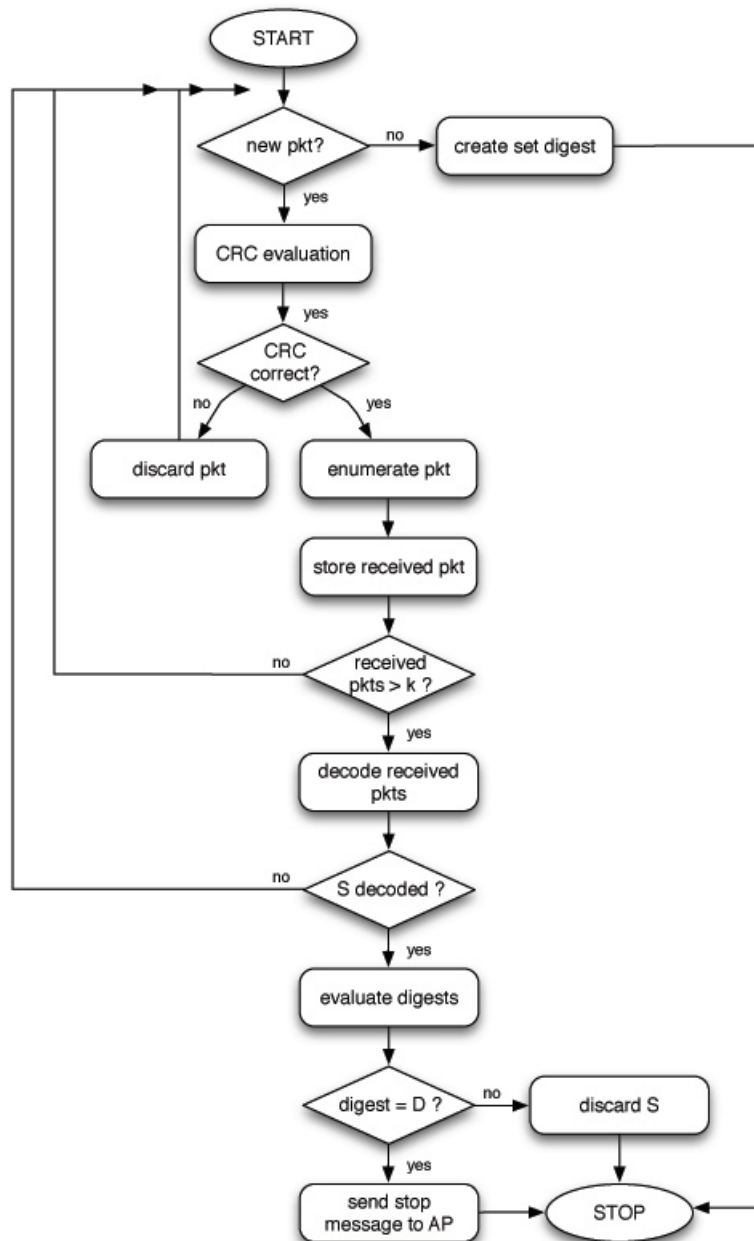


Figure 6.6: CORP Algorithm. The diagram shows the receiving phase of the CORP algorithm.

6.2 An application of fountain codes: CORP

sets Λ_a and Λ_b . Instead of understanding which packets can be received, the node can just require ESs belonging to sets different than the ones associated to S_a and S_b . The dissemination process has to follow four basic rules in order to be efficient and fast. The first rule is that a node that owns the whole information disseminates it by generating new disjointed sets of ESs and delivering them with unicast connections. This allows APs to increase the information availability by generating unique sets of ESs (i.e., generated from a new generation seed) per each new connection. By providing always

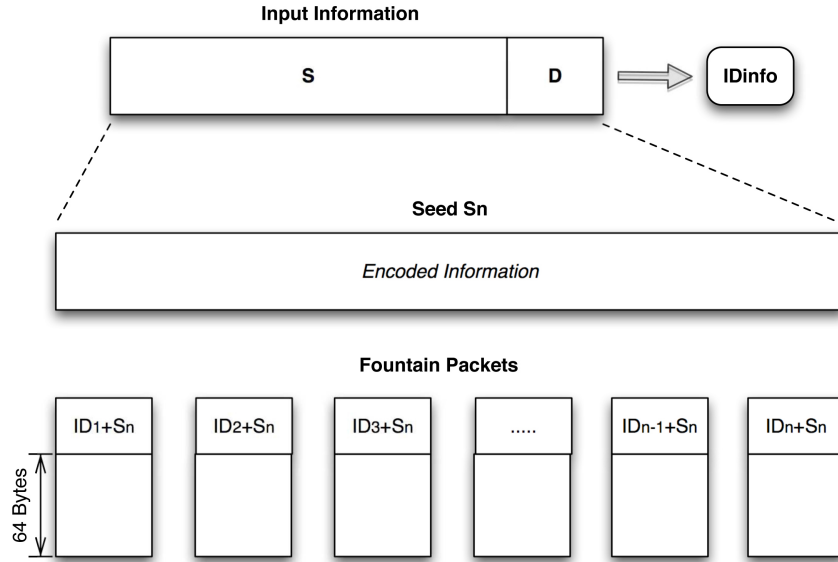


Figure 6.7: Coding in CORP. The figure shows the encoding symbols generation process in CORP protocol.

disjointed sets of ESs, the content reconciliation phase is simplified. In our approach, the use of unicast connections is more efficient than using broadcasting. In fact, despite its simplicity, broadcasting would not increase the information availability, and an expensive content reconciliation phase would be necessary.

The second rule we introduce is that a node cannot disseminate ESs belonging to the set it is receiving. This rule avoids the drawbacks of broadcast transmissions, restraining the uncontrolled spread of ESs that would yield to a long content reconciliation phase. For example, let assume that node A is receiving Λ_a from the AP and that the rule is not used. At a certain moment of the transfer, node B arrives close to node A and

6. DATA DISSEMINATION WITH RATELESS CODES

receives some of the ESs belonging to Λ_a before leaving. In this way, node B will have only a subset of Λ_a . If in the future node B want to receive the missing ESs belonging to that set, then it will have to perform the content reconciliation phase on the ESs.

A particular situation in the dissemination scenario is when a node can receive ESs belonging to a certain set from two or more neighbors. The scenario requires a rule to define how the downloading activity is performed.

The third rule is that a node cannot download information of a certain set of ESs by multiple nodes at the same time. According to this rule, a node can receive multiple sets at the same time only if these are disjointed. In this way, there is no need for coordination among the sending nodes. This is useful because it avoids the case in which nodes that are not within each others radio coverage serve the same requesting node with the same set. Although we stated that only one node can service a certain set, we still have to establish which of them will do it. In fact, because of poor channel conditions, nodes might not have received the same number of ESs belonging to that set. The fourth rule that we introduce is called certification strategy. According to this strategy, a node can disseminate a set only if it owns all the ESs of that set that are present on the network. Thus, a received set of ESs is said to be certified if the receiving node has indeed received all the ESs that are present on the network. For simplicity, we also say that a node is certified for a set if only if it owns all the ESs generated from a certain generation seed that are present in the network.

By using the certification strategy, we further simplify the content reconciliation strategy. In fact, the content reconciliation is performed only on certified sets. If the conditions of the channel are optimal, i.e., no packets are lost, then all sets will be certified. This would yield an optimal dissemination of the sets. However, when the channel conditions are poor, the certification strategy limits the transmission of sets only to nodes with enough information to deliver. Therefore, the strategy might yield a sub-optimal dissemination of the information. However, the content reconciliation time will be minimal, so that most of the connection time (that, in this case, would be very short) can be dedicated to data transmission. Moreover, even if a node cannot disseminate a set that is not certified, the ESs of that set are still useful for the decoding of the information block.

6.2.4 Packet definition

The CORP application packet is made up of an header and a payload. The header contains a field, *Type*, that indicates the payload type. The payload contains the data corresponding to the type indicated in the header and, optionally, a CRC to avoid data corruption. There are eight different types of packets:

1. *Beacon*

This packet is broadcasted by AP nodes to publish the list of the available information files. Each file is identified by a unique tag (4 bytes).

2. *Request*

This packet is broadcasted to request a specific file.

3. *Offer*

This packet is sent either by AP or MN nodes as a response to a request for a file. In this packet, the seeds S of the certified sets Λ and the parameters needed for the decoding process are specified. For example, it is indicated that the block is composed of 8000 source symbols of 64 bytes and that the adopted code is a LT code with given parameters δ and c . The packet has also a field that indicates if the offering node owns the whole file.

4. *Ack*

This packet is sent by the requesting node to communicate the set of ESs it wants to download by indicating the generating seed set. In addition, the requesting node can also specify the Starting Sequence Number (SSN), which is the first ES of the set that it wants to receive.

5. *Data*

This packet contains the encoded data and a Sequence Number (SN). The encoded data is a group of one or more ESs belonging to the same. SN indicates the number of the packet in the generated set of ESs.

6. *Stop*

This packet is sent by the requesting node to terminate the download of a certain set.

6. DATA DISSEMINATION WITH RATELESS CODES

7. *EoS*

This packet is sent by the offering node to communicate the end of the transmission of the set of ESs it is providing to a requesting node. The packet includes also a digest computed on the certified set, useful to the receiving node to check if the set can be tagged as certified.

6.2.5 Communications

The communication protocol is made up of two phases: the communication between AP and MN (Phase 1) and between two MNs (Phase 2). The information file is composed by the data information and a digest computed on the data file. The whole information file is then disseminated according to the CORP protocol. For simplicity, in the following we present the case where a file is composed of one block.

6.2.5.1 Phase 1: AP to MN

The AP periodically advertises the list of the available information files by broadcasting Beacon packets. If a MN within the radio coverage area of AP is interested in receiving one of the files, it broadcasts a Request packet specifying the relative tag. In particular, starting from this moment, the MN periodically sends a broadcast Request packet for the same file until it will complete the download.

When the AP receives the Request, it chooses a unique generation seed for the generation of the set of ESs to be transmitted. In addition, the packet also contains all the parameters that the rateless decoder needs to recover the information. After receiving the packet offer, the MN sends an ACK packet to confirm the settings of the AP for the transmission. Then, the AP can begin the data transmission and will continue until one of the following stop conditions occur:

- a) the MN is out of the radio coverage;
- b) the MN issues a stop packet.

If the MN realises that it is out of the APs radio coverage, then it assumes that the transmission is nished. As a consequence, if it has not received yet enough packets to decode, it computes a digest of the received ESs set and considers the set certified, because it has all the ESs present in the network. Starting from this moment, the MN

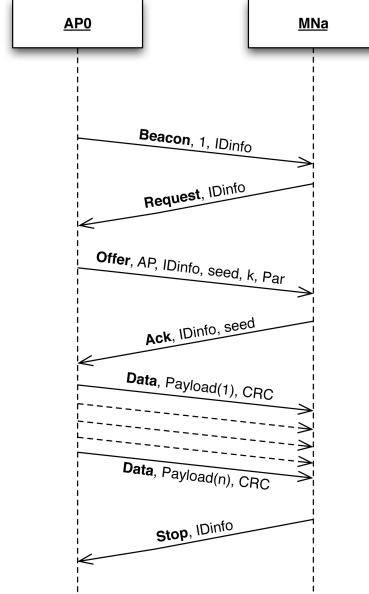


Figure 6.8: AP to MN communication. The figure shows the sequence diagram of the Access Point to Mobile Node communication.

can disseminate the received set of ESs.

If the same MN associates again to the AP and requests the same file, a new session will be opened and a new generation seed will be created. *In this way, the AP increases the availability of the information on network and there is no need for a content reconciliation phase.*

If the MN has received enough packets to decode, then it computes the digest of the recovered information and checks if it corresponds to the one present in the recovered file. If so, then the node has successfully received the source and, if still connected to the AP, then a Stop packet is generated. Moreover, the node will be able to provide the received file to other MNs generating new disjointed sets of ESs. However, if the digests do not match, then the data file is corrupted and will be immediately discarded.

6.2.5.2 Phase 2: MN to MN

In the dissemination between two MNs, there are two possibilities: The first one is when both nodes do not have enough ESs in order to decode the original information, the second one is when one of the nodes has the whole file. In the first case, once a MN

6. DATA DISSEMINATION WITH RATELESS CODES

receives a broadcast Request packet from another MN, it responds with a Offer packet containing the list of the generation seeds of the certified sets belonging to the requested file and specifying that it does not have the whole le. The requesting node will choose

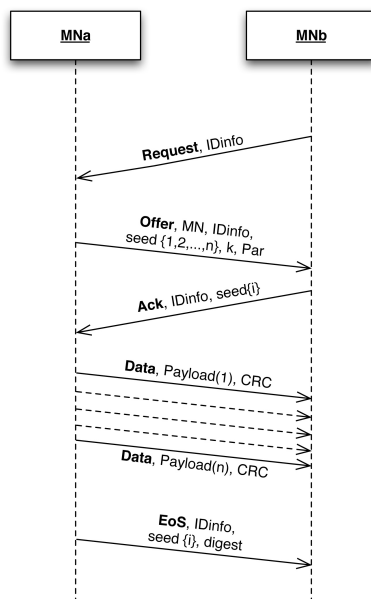


Figure 6.9: MN to MN communication. The figure shows the sequence diagram of the communication among mobile nodes in CORP.

one of the seeds and the sending node will start to transmit the ESs belonging to the set until one of the conditions presented in section 6.2.5.1 occur or if the sending node has sent all the ESs belonging to that set. In the latter case, the sending node transmits a EoS packet containing the digest of the provided set. At this point, the receiver, computes the digest of the set based on the ESs that it has received. If the computed digest matches the one contained in the EoS packet, then the set is certified and can be disseminated.

However, if there is no match between the two digests, then the node will not be able to disseminate the set but the ESs will be equally useful for the decoding of the information block. In the case where one node owns the whole information, this node answer to a Request by providing a unique generation seed and specifying that it owns the whole information. From this point the communication algorithm works exactly as specified in section 6.2.5.1.

6.2.6 CORP-TCP

CORP-TCP is a data dissemination protocol designed and implemented in order to evaluate CORP performances. In the next section we present the major outcome of the simulations carried out to compare CORP to I2V and V2V approaches. However, an additional comparison with a similar I2V2V was originally missing and CORP-TCP was what we really needed to highlight the benefits of our protocol.

The main difference between CORP and CORP-TCP is the transport protocol used at lower layer. In fact, while the former is developed on the top of UDP, the latter is completely designed to work on TCP. Clearly, by involving a complex transport layer such as TCP, the need of a coding scheme was not required anymore. According to CORP-TCP, the only information that has to be carried out is the fragmentation of the information S into smaller k fragments and the key idea behind this protocol is similar to the well-known P2P: if a vehicle moving in the simulation area is interested to a resource advertised by the AP, it simply start to download it. However, differently from CORP, it constantly keeps track of the received fragments so that it can ask later for the missing ones (see Figure 6.10). It is important to highlight that the missing fragments can be request either to an AP or another mobile node, in a I2V2V fashion.

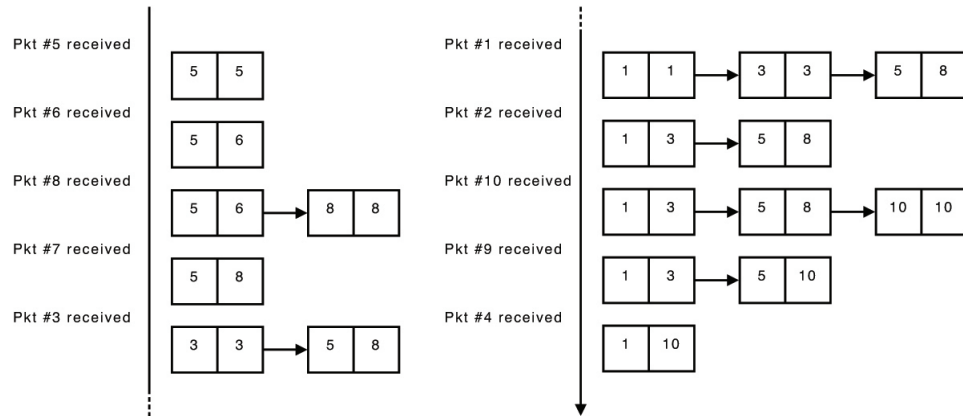


Figure 6.10: CORP-TCP. In CORP-TCP every mobile node has to keep track of the downloaded fragments. Each mobile node saves the first and the last fragment missing (that is the missing interval). The figure shows an example of a 10 fragments resource.

The packet structure is almost similar to the one used in CORP but there are still

6. DATA DISSEMINATION WITH RATELESS CODES

some important changes like the lack of the CRC field, which is not needed as such a checksum function is already provided by the TCP protocol. Another important change is the introduction of the *Connection table* and the *Service table*. The former is used to keep track of the connection with the other mobile nodes and it is made up of the following fields:

- *Connection ID*: the identifier of the connection with another vehicle;
- *Local Address*: the local interface IP address;
- *Remote Address*: the remote interface IP address related to a certain Connection ID;
- *Active*: this field is "true" if the connection is active, "false" otherwise.

An example of the Connection table is provided in Tab. 6.3.

Connection ID	Local Address	Remote Address	Active
1	192.168.0.10	192.168.0.11	true
2	192.168.0.10	192.168.0.14	false
⋮	⋮	⋮	⋮
n	192.168.0.10	192.168.0.32	true

Table 6.3: Example of a Connection table used in CORP-TCP.

The latter, instead, contains information related to the Services. An example of said table is provided in Tab. 6.4.

ID_{info}	Fragments list	Completed	Interesting	Requested
1	*list	true	true	true
2	*list	false	true	true
⋮	⋮	⋮	⋮	⋮
n	*list	false	true	true

Table 6.4: Example of a Service table used in CORP-TCP.

The contains the following fields:

- ID_{info} : the identifier of the service;

- *Fragments list*: the pointer to a sequenced list containing the sequence number of the received fragments;
- *Completed*: this field is "true" if the service has been successfully downloaded, "false" otherwise;
- *Interesting*: this field is "true" if the node is interested in that service, "false" otherwise;
- *Requested*: this field is "true" if the service has been successfully requested, "false" otherwise.

6.2.7 Simulations and results

In order to evaluate the performance of CORP, we modelled the system as a wireless ad hoc network with several MNs and a fixed AP. We considered the nodes to be equipped with omnidirectional radio transmitters with a nominal 216 meters radiorange and capable to reach a maximum bandwidth of 2Mbps. The transmission is UDP based and the performance of the system is simulated with QualNet network simulator. We assume that MNs can move freely (with speed ranging from 10 to 20 m/s and maximum pause-time of 15 seconds) within a 3000x3000 m wide area according to the RWP model. Instead, the AP is fixed and located in the center of the map in order to maximize the number of times a MN can communicate with it.

The application we considered for the simulation is the advertisement by a restaurant of a particular menu for lunch. The menu is contained in a 512KB image file that will be delivered in the said area to all the vehicles interested in that service. We performed 15 minutes simulations with a number of interested vehicles ranging from 2 to 100 and then we compared CORP to other two approaches. The first one is a basic scheme of information dissemination from the AP using rateless codes. This approach is not collaborative and vehicles can recover the information as soon as they have received enough ESs. We call this approach BRP. The second approach is DDRP, where nodes can exchange information only if they have successfully decoded the file. Among the rateless codes implementations presented in literature, in the simulations we decided to use LT codes (58) as they represent the first implementation of the digital

6. DATA DISSEMINATION WITH RATELESS CODES

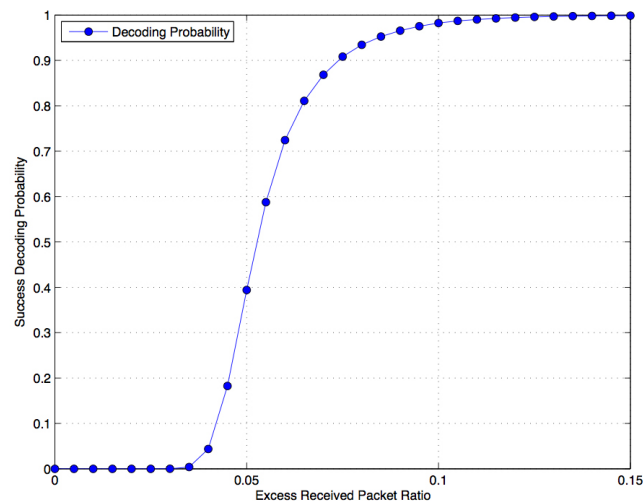


Figure 6.11: CORP. Cumulative success decoding probability distribution as a function of the excess received packets ratio with 8000 source packets of 64 Bytes.

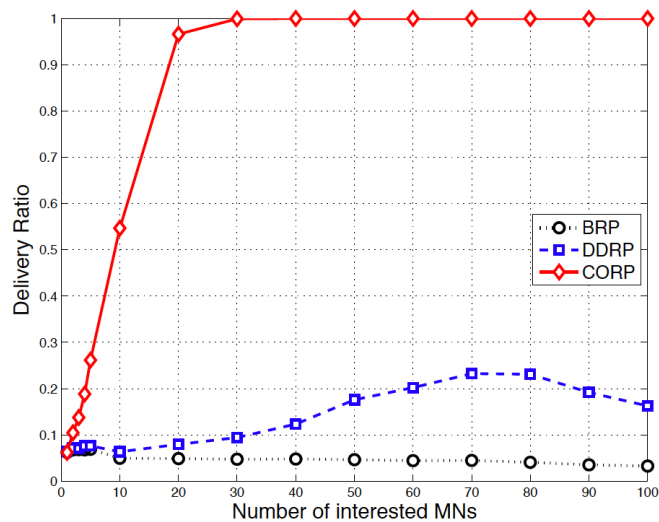


Figure 6.12: CORP delivery ratio. The figure shows the delivery ratio as a function of the number of cooperative MNs in the simulated area after 15 minutes.

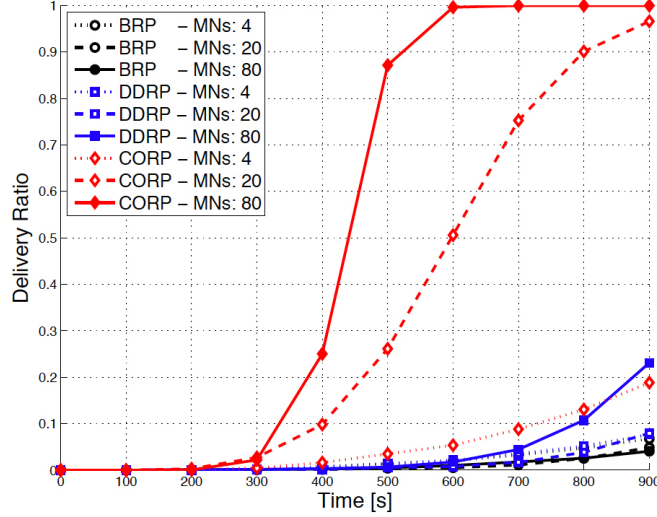


Figure 6.13: CORP delivery ratio. The figure shows the delivery ratio as a function of the time for different numbers of collaborating nodes. Three density scenarios are considered with respect to the number of mobile nodes: $MNs = 4$ (dotted lines), $MNs = 20$ (dashed lines) or $MNs = 80$ (continuous lines).

fountain coding mechanism.

The decoding performance of the adopted LT code is presented in Fig. 6.11. In this figure it is shown the probability of decoding as function of the extra packets ratio needed to recover the source information. As we can observe, most decoding processes succeed when the number of excess ESs is between 4% and 7%.

As it concerns the simulation parameters related to fountain coding are grouped in a quadruplet: $(K; \delta; c; l) = (8000; 0.5; 0.04; 64)$; the variable K represents the number of source packets of l bytes that form the information, while d and c are the parameters of the *Robust Soliton Degree Distribution*.

In order to cope with fading, multipath and high dynamism effects - which in vehicular networks can dramatically increase the bit error rate and, therefore, the packet loss rate (49) -, the dimension of data packets is rather small (64 bytes). As a consequence, by increasing reliability, the drawback is the decrease of efficiency due to the high packet overhead. the main simulation parameters are grouped in Tab.6.5 while Fig. 6.12 depicts the delivery ratio as a function of the number of cooperative MNs in the simulated area after 15 minutes. It is important to highlight that the delivery ratio is intended as the ratio of MNs which successfully decoded the information over the total

6. DATA DISSEMINATION WITH RATELESS CODES

number of interested vehicles in the area.

Fig. 6.12 shows that CORP performs better than the other approaches; while the CORP graph is monotonically increasing, the delivery ratio of the other two approaches is first growing and then it shows a decreasing trend.

With CORP, the collaboration of only 20 MNs is sufficient enough to recover the source information with a probability of 96%. Furthermore, delivery ratio is 1 even if the number of the interested nodes increases. On the other hand, when the DDRP approach is adopted, the average delivery ratio curve is increasing only when the number of MNs is lower than 70. However, also for few nodes, its cooperation policy is not satisfactory for the information reception. In particular, when more than 70 MNs are collaborating, the delivery ratio decreases. This happens because too many MNs are connected to the AP at the same time.

Parameter	Value
Simulation Time	900 s
Number of MNs	2 to 100
MN Mobility Model	RWP
MN Pause Time	15 [s]
MN Speed Range	10-20 [m/s]
Source Information K	8000 packets
Data Packet Payload Size	64 B
Initial Distribution MNs on map	Uniform
LT parameters: c, δ	0.04, 0.5
LT ES Size	64 B

Table 6.5: CORP simulation parameters used in Qualnet.

As a consequence, the amount of packets transmitted to each node from the AP decreases, thus requiring longer time to retrieve the file.

Finally, it is possible to take some consideration on BRP, which performs rather worse with respect to the others; in this case, the delivery ratio, which is almost constant up to 60 nodes, starts to decrease as the collaborating nodes increase, thus reducing the overall performance because the AP is required to handle too many connections at the same time. Clearly, this yields to a reduction of the bandwidth the AP allocates to each mobile connection. Furthermore, in the presence of parallel connections, the

channel's contention greatly reduce the available bandwidth, thus further decreasing the delivery ratio.

Fig. 6.13 shows the trend of the delivery ratio as a function of the time over the collaborating nodes. In this respect, three density scenarios are considered: 4 (dotted lines), 20 (dashed lines) and 80 (continuous lines) MNs. It is valuable to notice that the collaboration policy in CORP allows a major improvement of the delivery ratio with respect to the one used in DDRP. In particular, with only 4 MNs it is possible to reach the same performance of the DDRP solution with 80 MNs. Furthermore, with CORP, 10 minutes (simulation time) are enough to grant all the 80 MNs to receive and reassemble the relative information, while the same amount of time is completely inadequate in the other approaches. In Fig. 6.14 we present the average number of sets

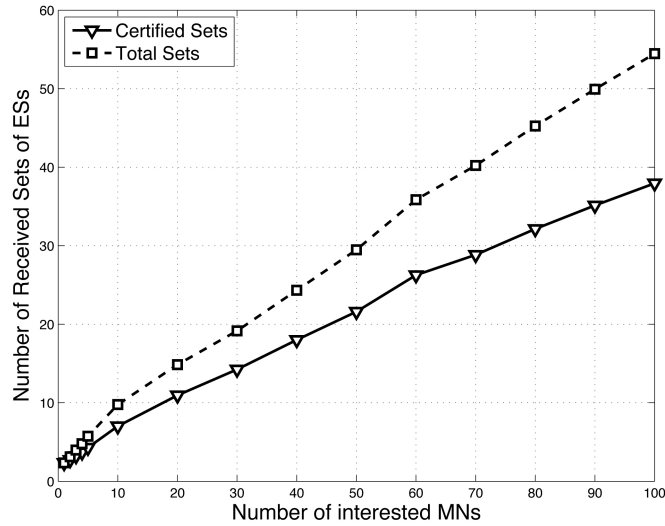


Figure 6.14: Number of Received Sets of ESs. The figure shows the total sets' number owned by each mobile node).

owned by each MN. In particular, the figure shows the total number of sets and the number of certified sets. As we can see, the trend for both graphs is linear. Moreover, the ratio between the number of certified sets and the total number of sets is almost constant.

Finally, in Fig. 6.15 we can see the average size of the shared ESs sets. In this case, the set size decreases as the number of nodes increases, which means that the AP is correctly allocating the available bandwidth among the connected MNs. However, this

6. DATA DISSEMINATION WITH RATELESS CODES

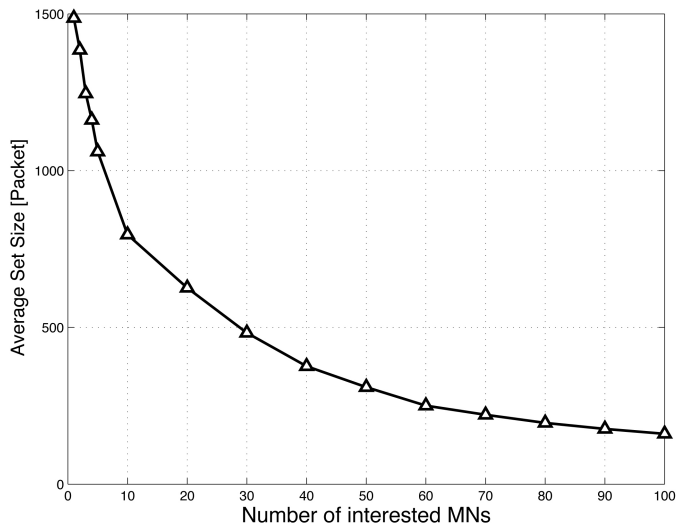


Figure 6.15: Average Set size in CORP. The figure shows the average set size over the number of mobile nodes. The set size decreases as the number of nodes increases, which improves the overall dissemination speed.

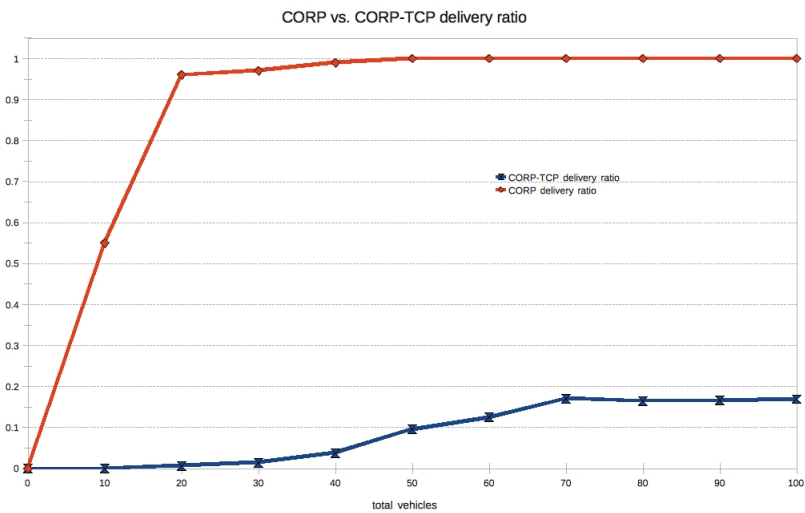


Figure 6.16: CORP vs. CORP-TCP delivery ratio. The figure shows the delivery ratio trend for CORP and CORP-TCP.



Figure 6.17: Aerial view of the simulation area. The figure shows the Google Earth aerial view of the simulation area. The related map in (Open Street Map) has been subsequently cropped and imported in Vanetmobisim in order to generate realistic mobility traces.

yields less delivered packets per connection, (i.e., the sets are smaller) but, on the other hand, this improve also the dissemination speed. In order to compare CORP to a similar I2V2V approach, we ran CORP-TCP simulations with the same scenario previously used with CORP and then we compared the delivery ratio of the two protocols. The comparison is shown in Fig. 6.16. The results show that CORP-TCP delivery ratio is clearly affected by the absence of any kind of coding and, which is most, by the TCP protocol with its intrinsic characteristics, namely, the three way handshake mechanism, the four-way connection closing mechanism and the slow start/restart mechanism.

The need of stronger and more accurate simulation results led us to test both CORP and CORP-TCP protocol in a pseudo-real scenario. As a consequence, we imported a real city map from *Open Street Map (OSM)*, which was subsequently cropped and exported as XML to Vanetmobisim(35) traffic simulator (fig. 6.19). In fig. 6.17 you can see the aerial view of the simulation area while in fig. 6.18 there is the related 2-D map. Once imported in Vanetmobisim, the map was then used to generate the realistic mobility patterns of the vehicles, taking into account both the speed limits, and the

6. DATA DISSEMINATION WITH RATELESS CODES

traffic lights (one per each intersection). The mobility model is the *Intelligent Driver Model with Lane changes* (IDM_{LC}(35), see section 4.2.4.1). This mobility model, extends the *IDM_{IM}* model (which, in turn adds intersection handling capabilities to the behavior of vehicles driven by the IDM scenarios) with the possibility for vehicles to change lane and overtake each others.

Parameter	Value
Simulation Time	900 s
Number of MNs	2 to 100
MN Mobility Model	IDM _{LC}
MN Speed Range	10-20 [m/s]
LT Source Information K	8000 packets
Data Packet Payload Size	64 B
Initial Distribution MNs on map	Uniform (constrained)
LT parameters: c, δ	0.04, 0.5
LT ES Size	64 B
Number APs:	1, in the middle of the map

Table 6.6: CORP and CORP-TCP simulation parameters used in Qualnet with the IDM_{LC} mobility pattern.

The parameters used for the simulation environment are shown in table 6.6. Results (fig. 6.21) show that CORP performs very well in the simulated scenario, and it really benefits from the FC coding. The difference in terms of delivery ratio between the two protocols prove, once again, that TCP intrinsic characteristics such as slow start-restart, three-way-handshake and four-way-disconnection are not suitable for a high dynamic scenario. With reference to fig. 6.21, please note that the delivery ratio starts decreasing when mobile nodes are more than 80-85. This effect is due to the high congestion caused by traffic lights. As a consequence, given that there is no active routing protocol, if vehicles stuck into the traffic the data dissemination process can be extremely difficult.

Furthermore, it is noticeable that CORP delivery ratio never reach 100% if compared to former results with RWP. This is because simulation has some minor flaws in mobility and few vehicles remain isolated and "far" from every other node in the area. Probably, this effect can be tricked with the use of straight maps.

6.2 An application of fountain codes: CORP



Figure 6.18: Imported map for mobility trace. The figure shows the map of the simulation area (it is a real map which includes two districts, "Statuario" and "Quarto Miglio").

Finally, we can say that the results registered by CORP are good and show that fountain code approach is suitable for data dissemination in vehicular ad-hoc networks.



Figure 6.19: Realistic mobility pattern. The figure shows the mobility pattern generated on a real map, according to the IDM.LC mobility model (The red dots represent the constrained vehicles moving across roadmap during the simulation.). This trace has been used to compare CORP and CORP-TCP performances.



Figure 6.20: Realistic mobility pattern - Detailed view. The figure shows the situation of the mobility pattern generated on a real map, according to the IDM.LC mobility model (The red dots represent the constrained vehicles moving across roadmap during the simulation.).

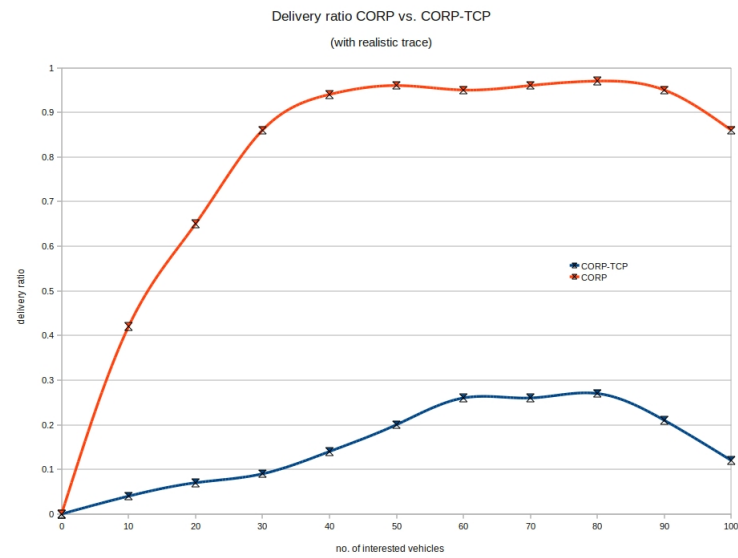


Figure 6.21: CORP vs. CORP-TCP delivery ratio with a realistic mobility model (IDM_{LC}). The figure shows the delivery ratio trend for CORP and CORP-TCP; simulations have been performed under a realistic mobility model such as the IDM_{LC}.

6. DATA DISSEMINATION WITH RATELESS CODES

Chapter 7

Conclusions

In this thesis we faced the problem of efficiently disseminate data in vehicular networks. Achieving this goal in a wireless environment usually involves a lot of challenging issues mainly due to the characteristics of the medium and the lack of synchronization and organization among nodes.

This is even more challenging in vehicular networks, as the nodes can move at high speed in a wide area surrounded of buildings and many other architectural structures which can, in turn, affect the propagation of the radio signal.

As a consequence, we introduced an innovative method based on rateless codes for a reliable and efficient data dissemination in vehicular networks: "*CORP*" (*Cooperative Rateless Protocol*). Our approach, which is designed on the top of UDP transport protocol, exploits the collaboration among vehicles in order to quickly disseminate an information on a wide area and represents the first successful application of the Fountain Codes theory to the vehicular field which totally complies with the *infrastructure-to-vehicle-to-vehicle* communication paradigm (I2V2V).

The validity of CORP was confirmed through extensive simulation with Qualnet 4.5, an excellent simulator which includes several libraries for accurate modeling of mobility and radio propagation effects.

The collected results show that CORP leads to an efficient dissemination in terms of both speed and reliability. In particular, the more vehicles cooperate, the faster the

7. CONCLUSIONS

dissemination. Furthermore, CORP performs better than other similar approaches, namely V2V, I2V and TCP based file sharing, even when only a limited number of vehicles cooperates. For example, 30 mobile nodes moving in a 3000x3000 m area (i.e. a car every 550 square meters on average) according to the RWP mobility model is enough to deliver an information among all the vehicles within 15 minutes. Slightly different is the situation in case of pseudo-real mobility trace (IDM-LC mobility model), where at least 50 cooperating nodes in a 1700x300 m area are needed in order to reach a delivery ratio of about 96%.

Additional measurements and analysis are being performing in order to investigate the impact of other relevant key parameters on the protocol, namely:

- the beaconing time for neighbors discovering;
- the frequency of the advertisement message sent by the AP to promote the service;
- the frequency of the periodic request packets;
- the influence of the transmission parameters such as power transmission;
- the packet size to be transmitted and the service size to be disseminated.

Future work will be mainly focused on the performance analysis of the protocol in real scenarios (i.e. rural, city, urban and freeway), with respect to the effects on signal transmission caused by architectural environment and surrounding buildings. In addition to simulation, the deployment of CORP protocol on a real vehicular testbed is currently being developed, thanks to the cooperation of University of Rome Tor Vergata, University of California Los Angeles (UCLA) and the Mobile Communications Department at EURECOM (France).

References

- [1] Canumobisim Vehicular Mobility Simulator Environment, available at <http://canu.informatik.uni-stuttgart.de/mobisim/>. 73, 77
- [2] Glomosim network simulator. 45, 73
- [3] The network simulator - NS2. 45, 58, 73
- [4] Qualnet network simulator. 45, 73
- [5] TSIS-CORSIM. 69
- [6] U.S. Census Bureau Tiger System Database. 73
- [7] *ASH: Application-aware SWANS with highway mobility*, 2008. 68, 93, 99
- [8] IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput. *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, pages c1–502, 29 2009. 22
- [9] RUDOLF AHLWEDE, NING CAI, SHUO YEN ROBERT LI, RAYMOND W. YEUNG, SENIOR MEMBER, AND SENIOR MEMBER. **Network information flow**. *IEEE Transactions on Information Theory*, **46**:1204–1216, 2000. 117, 118, 124
- [10] MOHIUDDIN AHMED, CEM U. SARAYDAR, TAMER ELBATT, JIJUN YIN, TIMOTHY TALTY, AND MICHAEL AMES. **Intra-vehicular Wireless Networks**. 28
- [11] O. ANDRISANO, R. VERDONE, AND M. NAKAGAWA. **Intelligent transportation systems: the role of third generation mobile radio networks**. *Communications Magazine, IEEE*, **38**(9):144–151, sep 2000. 36
- [12] A. BACHIR AND A. BENSLIMANE. **A multicast protocol in ad hoc networks inter-vehicle geocast**. In *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, **4**, pages 2456–2460 vol.4, April 2003. 101
- [13] RIMON BARR, ZYGMUNT J. HAAS, AND ROBERT VAN RENESSE. *Scalable Wireless Ad hoc Network Simulation*, chapter 19, pages 297–311. CRC Press, August 2005. 91
- [14] CHRISTIAN BETTSTETTER. **Smooth is better than sharp: a random mobility model for simulation of wireless networks**. In *MSWIM '01: Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 19–27, New York, NY, USA, 2001. ACM. 77

REFERENCES

- [15] JEAN-YVES LE BOUDEC AND MILAN VOJNOVIC. **The random trip model: stability, stationary regime, and perfect simulation.** *IEEE/ACM Trans. Netw.*, **14**(6):1153–1166, 2006. 55, 57
- [16] AZZEDINE BOUKERCHE, HORACIO A.B.F. OLIVEIRA, EDUARDO F. NAKAMURA, AND ANTONIO A.F. LOUREIRO. **Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems.** *Computer Communications*, **31**(12):2838 – 2849, 2008. Mobility Protocols for ITS/VANET. 12
- [17] JOHN W. BYERS, MICHAEL LUBY, MICHAEL MITZENMACHER, AND ASHUTOSH REGE. **A digital fountain approach to reliable distribution of bulk data.** In *SIGCOMM '98: Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 56–67, New York, NY, USA, 1998. ACM. 141
- [18] R. DIACONESCU V. CRISTEA C. GORGORIN, V. GRADINESCU AND L. IFODE. **An Integrated Vehicular and Network Simulator for Vehicular Ad-Hoc Networks.** Proceedings of the 20th European Simulation and Modelling Conference, October 2006. 59
- [19] P. CATALDI, A. TOMATIS, G. GRILLI, AND M. GERLA. **CORP: Cooperative rateless code protocol for vehicular content dissemination.** pages 1–7, 29 2009-july 1 2009. 1, 144
- [20] P. CATALDI, A. TOMATIS, G. GRILLI, AND M. GERLA. **A Novel Data Dissemination Method for Vehicular Networks with Rateless Codes.** In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6, April 2009. 115, 117
- [21] DAVID R. CHOFFNES AND FABIÁN E. BUSTAMANTE. **An integrated mobility and traffic model for vehicular wireless networks.** In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 69–78, New York, NY, USA, 2005. ACM. 58, 59, 94
- [22] F. DELAMARE, F.L. DOTTI, P. FERNANDES, C.M. NUNES, AND L.C. OST. **Analytical modeling of random waypoint mobility patterns.** In *Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks*, pages 106–113. ACM New York, NY, USA, 2006. 146
- [23] ROS G. DIMAKIS, VINOD PRABHAKARAN, AND KANNAN RAMCH. **Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes.** pages 111–117, 2005. 126
- [24] TAMER ELBATT, SIDDHARTHA K GOEL, GAVIN HOLL, HARIHARAN KRISHNAN, AND JAYENDRA PARIKH. **Cooperative Collision Warning Using Dedicated Short Range Wireless Communications.** In *VANET*, pages 1–9. ACM Press, 2006. 10
- [25] AHMED HELMY FAN BAI, NARAYANAN SADAGOPAN. **IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of Routing protocols for Adhoc NeTworks.** pages 825–835, 2003. 54, 55, 61, 64, 65
- [26] E. FASOLO, M. ROSSI, J. WIDMER, AND M. ZORZI. **In-network aggregation techniques for wireless sensor networks: a survey.** *Wireless Communications, IEEE*, **14**(2):70–87, April 2007. 116
- [27] JON FELDMAN, TAL MALKIN, ROCCO A. SERVEDIO, AND CLIFF STEIN. **Secure Network Coding via Filtered Secret Sharing.** 2004. 126
- [28] MARCO FIORE, CLAUDIO CASETTI, AND CARLA-FABIANA CHIASSERINI. **On-demand content delivery in vehicular wireless networks.** In *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 87–94, New York, NY, USA, 2005. ACM. 113

REFERENCES

-
- [29] CHRISTINA FRAGOULI, JEAN-YVES LE BOUDEDEC, AND JÖRG WIDMER. **Network coding: an instant primer.** *SIGCOMM Comput. Commun. Rev.*, **36**(1):63–68, 2006. 120
 - [30] CHRISTINA FRAGOULI, ATHINA MARKOPOULOU, ÉCOLE POLYTECHNIQUE, AND FÉDÉRALE LAUSANNE. **A network coding approach to overlay network monitoring.** In *In Allerton*, 2005. 126
 - [31] CHRISTINA FRAGOULI AND JÖRG WIDMER. **A Network Coding Approach to Energy Efficient Broadcasting: from Theory to Practice.** In *In Proc. of IEEE INFOCOM*. IEEE Computer Society, 2006. 125
 - [32] CHRISTINA FRAGOULI AND JÖRG WIDMER. **On the Benefits of Network Coding for Wireless Applications.** In *In 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–6, 2006. 125
 - [33] MARIO GERLA, KEN TANG, AND RAJIVE BAGRODIA. **TCP Performance in Wireless Multi-hop Networks.** pages 41–50, 1999. 129
 - [34] CHRISTOS GKANTSIDIS, JOHN MILLER, AND PABLO RODRIGUEZ. **Comprehensive view of a live network coding P2P system.** In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 177–188, New York, NY, USA, 2006. ACM. 126
 - [35] J. HÄRRI, F. FILALI, C. BONNET, AND MARCO FIORE. **VanetMobiSim: generating realistic mobility patterns for VANETs.** In *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 96–97, New York, NY, USA, 2006. ACM. 73, 146, 165, 166
 - [36] TRACEY HO, RALF KOETTER, MURIEL MÉDARD, DAVID R. KARGER, AND MICHELLE EFFROS. **The benefits of coding over routing in a randomized setting.** In *In Proceedings of 2003 IEEE International Symposium on Information Theory*, 2003. 123
 - [37] TRACEY HO, BEN LEONG, RALF KOETTER, MURIEL MEDARD, MICHELLE EROS, AND DAVID R. KARGER. **Byzantine Modification Detection in Multicast Networks using Randomized Network Coding.** In *in IEEE Proc. Intl. Sym. Inform. Theory*, page 143, 2004. 126
 - [38] B. HOH, M. GRUTESER, H. XIONG, AND A. ALRABADY. **Enhancing Security and Privacy in Traffic-Monitoring Systems.** *Pervasive Computing, IEEE*, **5**(4):38–46, oct.-dec. 2006. 36
 - [39] XIAOYAN HONG, MARIO GERLA, GUANGYU PEI, AND CHING-CHUAN CHIANG. **A group mobility model for ad hoc wireless networks.** In *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–60, New York, NY, USA, 1999. ACM. 50, 65
 - [40] XIAOYAN HONG, TAEK JIN KWON, MARIO GERLA, DANIEL LIHUI GU, AND GUANGYU PEI. **A Mobility Framework for Ad Hoc Wireless Networks.** In *MDM '01: Proceedings of the Second International Conference on Mobile Data Management*, pages 185–196, London, UK, 2001. Springer-Verlag. 65
 - [41] BARBARA HUGHES, RENÉ MEIER, RAYMOND CUNNINGHAM, AND VINNY CAHILL. **Towards real-time middleware for vehicular ad hoc networks.** In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 95–96, New York, NY, USA, 2004. ACM. 12
 - [42] International Symposium on Communications Internetworking. *Performance evaluation of TCP over UMTS transport channels Channels*, Ottawa - Canada, November/December 2004. 10
 - [43] J. BYERS, J. CONSIDINE, M. MITZENMACHER, S. ROST. **Informed Content Delivery Across Adaptive Overlay Networks.** In *ACM SIGCOMM 2002*, Pittsburgh - Pennsylvania - USA, 2002. 147

REFERENCES

- [44] DAVID B. JOHNSON AND DAVID A. MALTZ. **Dynamic source routing in ad hoc wireless networks.** In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996. 45, 57, 73
- [45] FRANK KARGL AND ELMAR SCHOCH. **Simulation of MANETs: a qualitative comparison between JIST/SWANS and ns-2.** In *MobiEval '07: Proceedings of the 1st international workshop on System evaluation for mobile platforms*, pages 41–46, New York, NY, USA, 2007. ACM. 91
- [46] F.K. KARNADI, ZHI HAI MO, AND KUN CHAN LAN. **Rapid Generation of Realistic Mobility Models for VANET.** *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 2506–2511, March 2007. 81
- [47] BRAD KARP AND H. T. KUNG. **GPSR: greedy perimeter stateless routing for wireless networks.** In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM. 40
- [48] WOLFGANG KIESS AND MARTIN MAUVE. **On the nature of Inter-Vehicle Communication**, February–March 2007. 36
- [49] J. KORHONEN AND Y. WANG. **Effect of packet size on loss rate and delay in wireless links.** 3, pages 1608 – 1613 Vol. 3, march 2005. 161
- [50] DEJAN KOSTIC, ADOLFO RODRIGUEZ, JEANNIE ALBRECHT, AND AMIN VAHDAT. **Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh.** In *In SOSp*, 2003. 147
- [51] STEFAN KRAUSS. *Microscopic Modelling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics.* PhD thesis, 1998. 80
- [52] JESSE LANDMAN AND PIETER KRITZINGER. **Delay analysis of downlink IP traffic on UMTS mobile networks.** *Performance Evaluation*, 62(1-4):68 – 82, 2005. Performance 2005. 13
- [53] ZHANG-W. FESTAG A. LE, L. AND R. BALDESSARI. **Analysis of Approaches for Channel Allocation in Car-to-Car Communication.** In *1st International Workshop on 1st International Workshop on Interoperable Vehicles*, March 2008. 32
- [54] FESTAG A. LE L., ZHANG W. AND BALDESSARI R. **Analysis of Approaches for Channel Allocation in Car-to-Car Communication Allocation in Car-to-Car Communication.** 1st International Workshop on Interoperable Vehicles, March 2008. 35
- [55] UICHIN LEE, JOON-SANG PARK, JOSEPH YEH, GIOVANNI PAU, AND MARIO GERLA. **Code torrent: content distribution using network coding in VANET.** In *MobiShare '06: Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking*, pages 1–5, New York, NY, USA, 2006. ACM. 113, 115
- [56] MINGHONG LIN, BIN FAN, JOHN C. S. LUI, AND DAH-MING CHIU. **Stochastic analysis of file-swarming systems.** *Perform. Eval.*, 64(9-12):856–875, 2007. 145
- [57] CHANGDONG LIU, M.J. LEE, AND T.N. SAADAWI. **A scalable multicast routing.** 2, pages 983 –987 vol.2, nov 1997. 40
- [58] M. LUBY. **LT codes.** *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 271–280, 2002. 136, 138, 148, 159
- [59] M. LUBY, M. WATSON, T. GASIBA, T. STOCKHAMMER, AND WEN XU. **Raptor codes for reliable download delivery in wireless broadcast systems.** *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, 1:192–197, Jan. 2006. 143

REFERENCES

- [60] P. MANZONI M. SANCHEZ. **A Java-based ad-hoc networks simulator**. *SCS Western Multiconference*, 1999. 44, 73
- [61] ATULYA MAHAJAN, NIRANJAN POTNIS, KARTIK GOPALAN, AND ANDY WANG. **Urban Mobility Models for VANETs**. In *Proc. of Workshop on Next generation Wireless Networks (WoNGen)*, December 2006. 53
- [62] C. MAIHOFFER. **A survey of geocast routing protocols**. *Communications Surveys Tutorials, IEEE*, 6(2):32–42, quarter 2004. 13
- [63] RAHUL MANGHARAM, DANIEL WELLER, RAJ RAJKUMAR, PRIYANTHA MUDALIGE, AND FAN BAI. **GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks (Invited Paper)**. 95
- [64] RAHUL MANGHARAM, DANIEL S. WELLER, DANIEL D. STANCIL, RAGUNATHAN RAJKUMAR, AND JAYENDRA S. PARIKH. **GrooveSim: A Topography-Accurate Simulator for Geographic Routing in Vehicular Networks**. In *in Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks*, pages 59–68, 2005. 95
- [65] B.M. MASINI, C. FONTANA, AND R. VERDONE. **Provision of an emergency warning service through GPRS: performance evaluation**. pages 1098 – 1102, oct. 2004. 36
- [66] P. MAYMOUNKOV. **Online Codes**. Tr2002-833, New York University, November 2002. 136, 148
- [67] PETAR MAYMOUNKOV AND DAVID MAZIERES. **Rateless Codes and Big Downloads**. In *In IPTPS'03*, 2003. 147
- [68] A. NANDAN, S. DAS, G. PAU, M. GERLA, AND M.Y. SANADIDI. **Co-operative downloading in vehicular ad-hoc wireless networks**. In *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on*, pages 32–41, Jan. 2005. 113, 114, 115, 127
- [69] VALERY NAUMOV, RAINER BAUMANN, AND THOMAS GROSS. **An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces**. In *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 108–119, New York, NY, USA, 2006. ACM. 11, 61
- [70] B. CLIFFORD NEUMAN. **Scale in Distributed Systems**. In *Readings in Distributed Computing Systems*, pages 463–489. IEEE Computer Society Press, 1994. 14
- [71] NHTSA. **Vehicle safety communications project-final report**. Technical Report HS 810 591, National Highway Traffic Safety Administration - NHTSA, April 2006. xv, 113
- [72] T. NOLTE, H. HANSSON, AND L.L. BELLO. **Automotive communications-past, current and future**. 1, pages 8 pp. –992, sept. 2005. 26
- [73] SANTASHIL PALCHAUDHURI, JEAN-YVES LE BOUDEC, AND MILAN VOJNOVIC. **Perfect Simulations for Random Trip Mobility Models**. In *ANSS '05: Proceedings of the 38th annual Symposium on Simulation*, pages 72–79, Washington, DC, USA, 2005. IEEE Computer Society. 90
- [74] GIANLUCA GRILLI ANDREA TOMATIS PASQUALE CATALDI, MARIO GERLA. **A novel data dissemination method for vehicular networks with rateless codes**. Technical Report TR090009, UCLA Computer Science Department, April 2009. 146
- [75] GIANLUCA GRILLI ANDREA TOMATIS PASQUALE CATALDI, MARIO GERLA. **Careless data dissemination in mobility environment with rateless codes**. Technical Report TR080023, UCLA Computer Science Department, July 2008. 146

REFERENCES

- [76] MICHAL PIORKOWSKI, MAXIM RAYA, ADA L. LUGO, PANOS PAPADIMITRATOS, MATTHIAS GROSSGLAUSER, AND JEAN-PIERRE HUBAUX. **TraNS: Realistic Joint Traffic and Network Simulator for VANETs**. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2007. 68, 96
- [77] NIRANJAN POTNIS AND ATULYA MAHAJAN. **Mobility models for vehicular ad hoc network simulations**. In *ACM-SE 44: Proceedings of the 44th annual Southeast regional conference*, pages 746–747, New York, NY, USA, 2006. ACM. 57, 58
- [78] BAI F. KENNEY J. RAI, V. AND K. LABERTEAUX. **Cross-Channel Interference Test Cross-Channel Interference Test Results: A Report from the VSC-A project**. Technical report, July 2007. 34
- [79] KENNEY J. RAI V., BAI F. AND LABERTEAUX K. **Cross-Channel Interference Test Results: A Report from the VSC-A project**. Technical report, July 2007. 35
- [80] KSCHISCHANG F. RALF KOETTER. **Coding for errors and erasures in random network coding**. In *in Proc. IEEE Int. Symp. Information Theory*, 2007. 147
- [81] C. L. ROBINSON, L. CAMINITI, D. CAVENEY, AND K. LABERTEAUX. **Efficient coordination and transmission of data for cooperative vehicular safety applications**. In *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 10–19, New York, NY, USA, 2006. ACM. 108
- [82] NARAYANAN SADAGOPAN, FAN BAI, BHASKAR KRISHNAMACHARI, AND AHMED HELMY. **PATHS: Analysis of PATH Duration Statistics and their Impact on Reactive MANET Routing Protocols**. In *in Proc. of MobiHoc*, pages 245–256, 2003. 65
- [83] P. SAENGUDOMLERT, E. MODIANO, AND R.G. GALLAGER. **On-line routing and wavelength assignment for dynamic traffic in WDM ring and torus networks**. *Networking, IEEE/ACM Transactions on*, 14(2):330–340, April 2006. 141
- [84] AMIT KUMAR SAHA AND DAVID B. JOHNSON. **Modeling mobility for vehicular ad-hoc networks**. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 91–92, New York, NY, USA, 2004. ACM. 57
- [85] PETER SANDERS, SEBASTIAN EGNER, AND LUDO TOLHUIZEN. **Polynomial Time Algorithms for Network Information Flow**. In *in 15th ACM Symposium on Parallel Algorithms and Architectures*, pages 286–294, 2003. 123, 124
- [86] H. SAWANT, JINDONG TAN, QINGYAN YANG, AND QIZHI WANG. **Using Bluetooth and sensor networks for intelligent transportation systems**. pages 767 – 772, oct. 2004. 26
- [87] THOMAS SCHULZE AND THOMAS FLIESS. **Urban traffic simulation with psycho-physical vehicle-following models**. In *WSC '97: Proceedings of the 29th conference on Winter simulation*, pages 1222–1229, Washington, DC, USA, 1997. IEEE Computer Society. 59
- [88] I. SESKAR, S.V. MARIC, J. HOLTZMAN, AND J. WASSERMAN. **Rate of location area updates in cellular systems**. In *Vehicular Technology Conference, 1992, IEEE 42nd*, pages 694–697 vol.2, May 1992. 77
- [89] A. SHOKROLLAHI. **Raptor codes**. *Information Theory, IEEE Transactions on*, 52(6):2551–2567, June 2006. 141, 142, 143
- [90] CHRISTOPH SOMMER, ZHENG YAO, REINHARD GERMAN, AND FALKO DRESSLER. **Simulating the Influence of IVC on Road Traffic using Bidirectionally Coupled Simulators**, 2008. 98
- [91] HENTY B. STANCIL D., CHENG L. AND BAI F. **Performance of 802.11p Waveforms over the Vehicle-to-Vehicle Channel at 5.9 GHz**. Technical report, September 2007. 35

REFERENCES

- [92] A. SUGIURA AND C. DERMAWAN. **In traffic jam IVC-RVC system for ITS using Bluetooth.** *Intelligent Transportation Systems, IEEE Transactions on*, **6**(3):302 – 313, sept. 2005. 26
- [93] V. DAVIES T. CAMP, J. BOLENG. **A survey of mobility models for ad hoc network research.** *Wireless Communications and Mobile Computing (WCMC)*, **2**(5):483–502, 2002. 43
- [94] M. TRANSIER W. EFFELSBERG T. KING, H FUSSLER. **Dead-reckoning for position-based forwarding on highways.** pages 199–204, Hamburg, Germany, March 2006. Proc. of the 3rd International Workshop on Intelligent Transportation (WIT 2006). 12
- [95] STATUS OF THIS, H. IZUMIYAMA, N. FUJII, AND Y. ZHANG. **A Link-Layer Tunneling Mechanism for Unidirectional Links**, 2001. 37
- [96] JING TIAN, J. HAHNER, C. BECKER, I. STEPANOV, AND K. ROTHERMEL. **Graph-based mobility model for mobile ad hoc network simulation.** In *Simulation Symposium, 2002. Proceedings. 35th Annual*, pages 337–344, April 2002. 77
- [97] M. TREIBER AND D. HELBING. **Realistische Mikrosimulation von Straßenverkehr mit einem einfachen Modell.** In *16th Symposium Simulationstechnik ASIM 2002*, 2002. 80
- [98] MARTIN TREIBER, ANSGAR HENNECKE, AND DIRK HELBING. **Congested Traffic States in Empirical Observations and Microscopic Simulations.** *Physical Review E*, **62**:1805, 2000. 73, 77
- [99] S. Y. WANG, C. L. CHOU, Y. H. CHIU, Y. S. TZENG, M. S. HSU, Y. W. CHENG, W. L. LIU, AND T. W. HO. **NCTUns 4.0: An Integrated Simulation Platform for Vehicular Traffic.** In *Communication, and Network Researches," WiVec'07*, 2007. 93, 98
- [100] AXEL WEGENER, MICHAŁ PIÓRKOWSKI, MAXIM RAYA, HORST HELLBRÜCK, STEFAN FISCHER, AND JEAN-PIERRE HUBAUX. **TraCI: an interface for coupling road traffic and network simulators.** In *CNS '08: Proceedings of the 11th communications and networking simulation symposium*, pages 155–163, New York, NY, USA, 2008. ACM. 97
- [101] T. WEIL. **Service Management for ITS Using WAVE (1609.3) Networking.** pages 1 –6, 30 2009-dec. 4 2009. 29, 34
- [102] CHRISTIAN WEWETZER, MURAT CALISKAN, KLAUS MEIER, AND ANDREAS LUEBKE. **Experimental Evaluation of UMTS and Wireless LAN for Inter-Vehicle Communication.** International Conference ITS Telecommunications, Sophia Antipolis, France, 2007. 10
- [103] WIKIPEDIA. **European Telecommunications Standards Institute — Wikipedia, The Free Encyclopedia**, 2010. [Online; accessed 28-February-2010]. 32
- [104] C. WU AND B. LI. **rStream: resilient peer-to-peer streaming with rateless codes.** In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 307–310. ACM New York, NY, USA, 2005. 146, 147
- [105] HAO WU, R. FUJIMOTO, AND G. RILEY. **Analytical models for information propagation in vehicle-to-vehicle networks.** In *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, **6**, pages 4548–4552 Vol. 6, Sept. 2004. 114
- [106] YUNNAN WU. **Information exchange in wireless networks with network coding and physical-layer broadcast.** 2004. 125
- [107] YUNNAN WU, PHILIP A. CHOU, AND KAMAL JAIN. **A Comparison of Network Coding and Tree Packing.** In *IN PROC. 2004 IEEE INTERNATIONAL SYMPOSIUM ON INFORMATION THEORY (ISIT 2004*, page 143, 2004. 123

REFERENCES

- [108] BO XU, A. OUKSEL, AND O. WOLFSON. **Opportunistic resource exchange in inter-vehicle ad-hoc networks**. In *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on*, pages 4–12, 2004. 114
- [109] QING XU, TONY MAK, JEFF KO, AND RAJA SENGUPTA. **Vehicle-to-vehicle safety messaging in DSRC**. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 19–28, New York, NY, USA, 2004. ACM. 107, 110, 114
- [110] MING YANG KUMARAWADU P. GHENNIWA H.H. WEIMING SHEN YUNJIAO XUE, HO SUNG LEE. **Performance Evaluation of NS-2 Simulator for Wireless Sensor Networks**. pages 1372 –1375, April 2007. 103
- [111] PHILIP CHOU YUNNAN, PHILIP A. CHOU, YUNNAN WU, AND KAMAL JAIN. **Practical Network Coding**, 2003. 122
- [112] XIANG ZENG, RAJIVE BAGRODIA, AND MARIO GERLA. **GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks**. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulation (PADS'98)*, pages 154–161. IEEE Computer Society, May 1998. 83
- [113] J. ZHAO AND G. CAO. **VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks**. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006. 114
- [114] MING ZHAO AND WENYE WANG. **WSN03-4: A Novel Semi-Markov Smooth Mobility Model for Mobile Ad Hoc Networks**. In *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pages 1–5, 27 2006-Dec. 1 2006. 90
- [115] QUNWEI ZHENG, XIAOYAN HONG, AND SIBABRATA RAY. **Recent advances in mobility modeling for mobile ad hoc network research**. In *ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference*, pages 70–75, New York, NY, USA, 2004. ACM. 43
- [116] M.M. ZONOOZI AND P. DASSANAYAKE. **User mobility modeling and characterization of mobility patterns**. *Selected Areas in Communications, IEEE Journal on*, **15**(7):1239–1252, Sep 1997. 44

LIST OF ACRONYMS

AMPS	Advanced Mobile Phone System
AODV	Ad-Hoc On demand Distance Vector routing
AP	Access Point
CAR-2-X	car to car + car to infrastructure communication
CBR	Constant Bit Rate
CCH	Control CHannel
CDMA	Code Division Multiple Access
CEPT	European Conference of Postal and Telecommunications Administrations
CSMM	City Section Mobility Model
C2C-C	Car-to-Car Communication
DCF	Distributed Coordinated Function
DF	Digital Fountain
DSDV	Destination-Sequenced Distance-Vector routing
DSR	Dynamic Source Routing
DSRC	Dedicated Short-Range Communications
EBW	Emergency Brake Warning
ECUs	Electronic Control Units
EDCA	Enhanced Distributed Channel Access
EIRP	Equivalent Isotropically Radiated Power
ES	Encoded Set
ETACS	Enhanced Total Access Communication System
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FDM	Frequency-Division Multiplexing
FHSS	Frequency-Hopping Spread Spectrum
FTM	Fluid Traffic Model
GDF	Geographical Data Files
GPS	Global Positioning System

REFERENCES

GPSR Greedy Perimeter Stateless Protocol

GSM Global System for Mobile Communications

GUI Graphical User Interface

IDM Intelligent Drive Model

IDM-LC Intelligent Drive Model - Lane Change

IEEE Institute of Electrical and Electronic Engineers

ITM Irregular Terrain Model

ITS Intelligent Transportation System

I2V Infrastructure to Vehicle

I2V2V Infrastructure to Vehicle to Vehicle

I2I Infrastructure to Infrastructure

InV In-Vehicle Communications

IP Internet Protocol

IVG Inter-Vehicle Geocast protocol

IVW Intersection Violation Warning

JTAC Japanese Total Access Communication

LAN Local Area Network

LDGM Low Density Generator Matrix

LDPC Low Density Parity Check

MAC Media Access Control

MAN Metropolitan Area Networks

MANET Mobile Ad-Hoc Networks

MDMM Modified Direction Mobility Model

MIMO Multiple-Input Multiple-Output

MM Mobility Model

MN Mobile Node

MOE Measure Of Effectiveness

NMT Nordic Mobile Telephone

OBU On Board data Unit

OEM Original Equipment Manufacturer

REFERENCES

OFDM Orthogonal frequency-division multiplexing

PER Packet Error Rate

PSDU Physical layer Service Data Unit

P2M Point to Multipoint

P2P Peer to Peer

QoS Quality of Service

RF Radio Frequency

RME Route Management and Execution

RPGM Reference Point Group Mobility Model

RSE Radio Signal Emitter or Radio Sensing Environment

RSU Road Side Unit

RTT Round Trip Trime

RUM Rice University Model

RWMM Random Walk Mobility Model

RWP Random Waypoint Model

SCH Service CHannel

SDM Spatial Division Multiplexing

SIG Special Interest Group

SSM Stop Sign Model

UCLA University of California Los Angeles

UMTS Universal Mobile Telecommunications System

UWB Ultra-Wide Band

TACS Total Access Communication System

TCP Transfer Control Protocol

TMC Traffic Message Channel

TSB Topologically Scoped Broadcast

TSM Traffic Sign Model

UDP User Datagram Protocol

UDLR UniDirectional Link Routing

VANET Vehicular Ad-Hoc Networks

REFERENCES

VBR Variable Bit Rate

VOIP Voice Over Internet Protocol

VSCC Vehicle Safety Communications Consortium

V2I Vehicle to Infrastructure

V2V Vehicle to Vehicle

WAVE Wireless Access Vehicular Environment

WHO World Health Organization

WiFi Wireless Fidelity

WLAN Wireless Local Area Networks

WiMAX Worldwide Interoperability for Microwave Access