UNIVERSITÀ DI PISA

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA SPECIALISTICA IN INGEGNERIA INFORMATICA

TESI DI LAUREA SPECIALISTICA

Vehicular Networks

Traffic Simulations and Communication Protocols

Relatori:

Prof. Paolo Ancilotti
Scuola Superiore Sant'Anna

Prof. Pierfrancesco Foglia
Dipartimento di Ingegneria dell'Informazione

Dott. Enrico Bini
Scuola Superiore Sant'Anna

Candidato:

Stefano Fontanelli

ANNO ACCADEMICO 2007/2008

Ai miei genitori
che hanno vissuto con me le gioie e i dolori di questi anni

Abstract

Traffic congestion wastes time, energy, and causes pollution. Recently, it has been estimated that congestion *produces* 3.5 billions hours delay and 5.7 billions gallons of wasted fuel every year in the U.S. Also, traffic congestion does not allow an efficient usage of the urban roadways. Often the shortest path is not the fastest one in metropolitan areas, therefore modern GPS navigators in addition to give wrong fastest path, increase traffic congestion because they direct vehicles along the same path. One possible remedy to the congestion is the exploitation of the vehicle-to-vehicle communication.

Vehicles equipped with devices capable of short-range wireless connectivity can form a particular mobile ad-hoc network, called a "Vehicular Ad-hoc NETwork" (VANET). The existence of such networks opens the way for a large range of applications. One of the most important classes of such applications is related to route planning. Route planning aims to provide drivers with real-time traffic information, which would require expensive infrastructure, in the absence of a VANET.

The evaluation of VANET protocols and applications could be made through real outdoor experiments, which should involve a large number of nodes, in order to obtain significant results. However, performing such large-scale experiments is extremely difficult and expensive. Therefore, simulation is an indispensable tool.

The simulation of the VANETs requires two different components: a vehicular traffic simulator, capable to provide an accurate mobility model for the nodes of a VANET, and a network simulator, for simulating the behavior of a wireless network. Recent studies have proven that the vehicular mobility model is very important, and it should be well integrated with the wireless network model, in order to obtain relevant results. The usage of an inaccurate mobility model, like the popular random waypoint model (which may work for some mobile ad-hoc networks, but is definitely not an accurate representation of mobility in a VANET), can lead to erroneous results. In order to study traffic congestion, combining an existing vehicular traffic simulator with an existing wireless network simulator is not possible. Hence, an integrated simulator is needed.

In the proposed work we addressed two goals: (1) we developed an integrated simulator, that can mix vehicular traffic simulator and selected wireless network simulator characteristics, and (2) we investigate the behaviour of some basic communication protocols to avoid traffic congestion. Finally, the communication protocols are extensively compared by their capability of reducing traffic congestion.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In last decades the Information Technology has advanced significantly, allowing new applications that were unrealistic before. The technology advance has been so dramatic that many easily exploitable applications have remained unexplored.

1.1 The wireless ad-hoc networks

A wireless ad-hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such an environment, it may be necessary for one mobile host to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile hosts wireless transmissions.

A Mobile Ad-hoc NETwork (MANET) is a collection of wireless nodes communicating with each other in the absence of any infrastructure. Due to the availability of small and inexpensive wireless communicating devices, the MANET research field has attracted a lot of attention from academia and industry in the recent years. In the near future, MANETs could potentially be used in various applications such as mobile classrooms, battlefield communication and disaster relief applications [13].

Vehicular Ad-hoc Networks (VANETs) represent a rapidly emerging, particularly challenging class of Mobile Ad Hoc Networks (MANETs).

VANETs are distributed, self-organizing communication networks built up by moving vehicles, and are thus characterized by a very high node mobility and limited degrees of freedom in the mobility patterns. Such particular features often make standard networking protocols inefficient or unusable in VANETs, whence the growing effort in the development of communication protocols which are specific to vehicular networks.

While it is crucial to test and evaluate protocol implementations in a real testbed environment, simulation is widely considered as a first step in the development of protocols as well as in the validation and refinement of analytical models for VANETs.

One of the critical aspects [41] when simulating VANETs is the employment of mobility models that reflect as closely as possible the real behavior of vehicular traffic. This notwithstanding, using simple random-pattern, graph-constrained mobility models is a common practice among researchers working on VANETs. There is no need to say that such models cannot describe vehicular mobility in a realistic way, since they ignore the peculiar aspects of vehicular traffic, such as cars acceleration and deceleration in presence of nearby vehicles, queuing at roads intersections, traffic bursts caused by traffic lights, and traffic congestion or traffic jams. All these situations greatly affect the network performance, since they act on network connectivity, and this makes the adoption of a realistic mobility model fundamental when studying VANETs.

1.2 The Mobility Models in Ad-hoc Networks

The mobility model is designed to describe the movement pattern of mobile users, and how their location, velocity and acceleration change over time. Since mobility patterns may play a significant role in determining the protocol performance, it is desirable for mobility models to emulate the movement pattern of targeted real life applications in a reasonable way. Otherwise, the observations made and the conclusions drawn from the simulation studies may be misleading.

In [13, 62] are discussed seven different synthetic entity mobility models for ad hoc networks that are usually used in literature:

1. Random Walk Mobility Model (including its many derivatives): A simple mobility model based on random directions and speeds.

2. Random Waypoint Mobility Model: A model that includes pause times between changes in destination and speed.

3. Random Direction Mobility Model: A model that forces MNs to travel to the edge of the simulation area before changing direction and speed.

4. A Boundless Simulation Area Mobility Model: A model that converts a 2D rectangular simulation area into a torus-shaped simulation area.

5. Gauss-Markov Mobility Model: A model that uses one tuning parameter to vary the degree of randomness in the mobility pattern.

6. A Probabilistic Version of the Random Walk Mobility Model: A model that utilizes a set of probabilities to determine the next position of an MN.

7. City Section Mobility Model: A simulation area that represents streets within a city.

In the next section we will describe the first and the second mobility model that are widely used in the simulations of the VANETs.

1.2.1 Random-based mobility models

In random-based mobility models, the mobile nodes move randomly and freely without restrictions. To be more specific, the destination, speed and direction are all chosen randomly and independently of other nodes. This kind of model has been used in many simulation studies.

The Random Waypoint Model in which nodes move independently to a randomly chosen destination with a randomly selected velocity. It was first proposed by Johnson and Maltz [37]. Soon, it became a 'benchmark' mobility model to evaluate the MANET routing protocols, because of its simplicity and wide availability.

In the random way-point model, a node starts at an initial location and moves towards a randomly chosen destination location on the simulation surface with a constant speed. The speed of motion is randomly chosen and the trajectory is a straight line. When it reaches its destination it pauses for some time and then chooses another destination and a random speed and moves towards it. It repeats this process throughout the simulation.

The simplicity of Random Waypoint model may have been one reason for its widespread use in simulations.

The Random Walk model was originally proposed to emulate the unpredictable movement of particles in physics. It is also referred to as the Brownian Motion. Because some mobile nodes are believed to move in an unexpected way, Random Walk mobility model is proposed to mimic their movement behavior [62]. The Random Walk model has similarities with the Random Waypoint model because the node movement has strong randomness in both models. We can think the Random Walk model as the specific Random Waypoint model with zero pause time.

However, in the Random Walk model, the nodes change their speed and direction at each time interval. For every new interval t, each node randomly and uniformly chooses its new direction $\theta(t)$ from $(0, 2\pi]$. In similar way, the new speed $v(t)$ follows a uniform distribution or a Guassian distribution from [0, V_{max}]. Therefore, during time interval t, the node moves with the velocity vector $(v(t)\ cos\theta(t),\ v(t)\ sin\theta(t))$. If the node moves according to the above rules and reaches the boundary of simulation field, the leaving node is bounced back to the simulation field with the angle of $\theta(t)$ or $\pi - \theta(t)$, respectively. This effect is called border effect [15].

The Random Walk model is a memoryless mobility process where the information about the previous status is not used for the future decision.

That is to say, the current velocity is independent with its previous velocity and the future velocity is also independent with its current velocity. However, we observe that is not the case of mobile nodes in many real life applications

1.3 Realistic Vehicular Mobility Models

In the literature, vehicular mobility models are usually classified as either microscopic or macroscopic.

- When focusing on a macroscopic point of view, motion constraints such as roads, streets, crossroads, and traffic lights are considered. Also, the generation of vehicular traffic such as traffic density, traffic flows, and initial vehicle distributions are defined.
- The microscopic approach, instead, focuses on the movement of each individual vehicle and on the vehicle behavior with respect to others.

Yet, this micro-macro approach is more a way to analyze a mobility model than a formal description.

Another way to look at mobility models is to identify two functional blocks: Motion Constraints and Traffic Generator. Motion Constraints describe how each vehicle moves (its relative degree of freedom), and is usually obtained from a topological map. Macroscopically, motion constraints are streets or buildings, but microscopically, constraints are modeled by neighboring cars, pedestrians, or by limited roads diversities either due to the type of cars or to drivers habits.

The Traffic Generator, on the other hand, generates different kinds of cars, and deals with their interactions according to the environment under study. Macroscopically, it models traffic densities or traffic flows, while microscopically, it deals with properties like inter-distances between cars, acceleration or braking.

A realistic mobility model should include:

1. Accurate and Realistic topological maps: Such maps should manage different densities of roads, contains multiple lanes, different categories

of streets and associated velocities.

2. Smooth deceleration and acceleration: Since vehicles do not abruptly break and move, deceleration and acceleration models should be considered.

3. Obstacles: We require obstacles in the large sense of the term, including both mobility and wireless communication obstacles.

4. Attraction points: As any driver knows, initial and final destination are anything but random. And most of the time, drivers are all driving in similar final destinations, which creates bottlenecks. So macroscopically speaking, drivers move between a repulsion point towards an attraction point using a drivers preferred path.

5. Simulation time: Traffic density is not uniformly spread around the day. An heterogeneous traffic density is always observed at some peak time of days, such as Rush hours or Special Events.

6. Non-random distribution of vehicles: As it can be observed in real life, cars initial positions cannot be uniformly distributed in a simulation area, even between attraction points. Actually, depending of the Time configuration, the density of cars at particular centers of interest, such as homes, offices, shopping malls are preferred.

7. Intelligent Driving Patterns: Drivers interact with their environments, not only with respect to static obstacles, but also to dynamic obstacles, such as neighboring cars and pedestrians. Accordingly, the mobility model should control vehicles mutual interactions such as overtaking, traffic jam, preferred paths, or preventive action when confronted to pedestrians.

1.4 VANETs Mobility Models

When mobility was first taken into account in simulation of wireless networks, several models to generate mobility patterns of nodes were proposed. The

Random Waypoint model, the Random Walk model, the Reference Point Group (or Platoon) model, the Node Following mode, the Gauss-Markov model, just to cite the most known ones, all involved generation of random linear speed-constant movements within the topology boundaries. Further works added pause times, reflection on boundaries, acceleration and deceleration of nodes. Simplicity of use conferred success to the Random Waypoint model in particular, however, the intrinsic nature of such mobility models may produce unrealistic movement patterns when compared to some real world behavior.

As far as Vehicular Ad-hoc Networks (VANETs) are concerned, it soon became clear that using any of the aforementioned models would produce completely useless results. Consequently, the research community started to seek more realistic models.

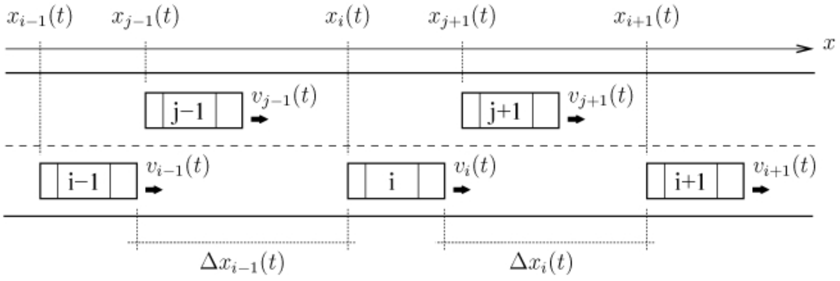

Figure 1.1: Vehicular traffic notation.

1.4.1 Notation

With reference to the vehicular traffic scenario depicted in Figure 1.1, we define as i the vehicle whose behavior is currently under investigation. At a given time instant t, such vehicle is at a position $x_i(t)$, and travels with a speed $v_i(t)$, meaning that its instantaneous acceleration can be expressed as $\frac{dv_i(t)}{dt}$. Indexes $i-1$ and $i+1$ identify the back and front vehicles with respect to i, which are located at $x_{i-1}(t)$ and $x_{i+1}(t)$, and travel at velocity $v_{i-1}(t)$ and $v_{i+1}(t)$, respectively, at time t. The front bumper to back bumper distance between i and $i+1$ is identified as $\Delta x_i(t)$. Indexes $j-1$ and $j+1$ identify the back and front vehicles with respect to i, in a different lane, considered for lane change. These vehicles are at positions $x_{j-1}(t)$ and $x_{j+1}(t)$ and travel with speed $v_{j1}(t)$ and $v_{j+1}(t)$, respectively. Also, we denote input parameters for the vehicular mobility descriptions with consistent notation in different models. The symbols used are listed in Table 1.1.

Parameter	Symbol
Acceleration	a
Deceleration	b
Drivers reaction time	τ
Time step	Δt
Maximum (desired) speed	v_{max}
Minimum speed	v_{min}
Safe time headway	T

Table 1.1: List of symbols

1.4.2 Stochastic models

The stochastic models are all those mobility descriptions which constrain random movements of nodes on a graph. The graph represents a road topology, and the movement is random in a sense that vehicles, individually or with group dynamics, follow casual paths over the graph, usually traveling at randomly chosen speed.

Stochastic models are the simplest vehicular mobility descriptions used in VANETs research, as they do not consider any vehicular traffic theory result. Basic aspects such as car-to-car interaction and intersection modeling are neglect or tackled simplistically, with the result that fundamental phenomena of vehicular mobility cannot be reproduced. Stochastic models are often compared against fully random mobility models, i.e., models that do not constrain the random nodes movement over a graph, such as the Random Walk [27] or the Random Waypoint [21]. However, such tests can hardly validate the realism of stochastic models.

The City Section mobility model introduced by Davies [25] constrains nodes movement on a grid road topology, where all edges are considered bi-directional, single-lane roads. Vehicles randomly select one of the intersections as their destination over the grid and move towards it with constant speed, with (at most) one horizontal and one vertical movement. The speed depends on the road the vehicle is traveling on: two road classes, high-speed and low-speed, are considered, and each node sets its speed to a high or low value accordingly. Car-to-car interactions are ignored, since all adjacent vehicles travel at the same speed, and vehicles are allowed to overlap at road junctions. No pauses at intersections, nor at the end of trips are specified.

The Freeway model representing several bi-directional multi-lane freeways. Within the IMPORTANT framework [31], Bai *et al.* adopt a Freeway mobility model and a Manhattan mobility model to verify the effect of different mobility descriptions on several metrics of networking interest. The movement of each node is restricted to the lane it is moving on, and the following speed rules apply to vehicle i:

1. $v_i(t + \Delta t) = v_i(t) + \eta a \Delta t$;
2. IF $v_i(t) < v_{min}$ THEN $v_i(t) = v_{min}$;
3. IF $v_i(t) > v_{max}$ THEN $v_i(t) = v_{max}$;
4. IF $\Delta x_i(t) \leq SD$ THEN $v_i(t) = v_{i+1}(t) - \frac{a}{2}$;

where SD is the minimum safety distance between two vehicles and η is a random variable uniformly distributed in $[-1,\ 1]$. The first rule adds some randomness to the update of vehicles speed, which is first selected as uniformly distributed in an interval $[v_{min},\ v_{max}]$. The second rule introduces a minimum safety distance requirement between vehicles. Each vehicle starts its movement at the beginning of a lane, and ends it once it reaches the end of the same lane. Then a new movement, on a randomly selected lane, is started over.

The Manhattan mobility model uses a grid road topology, similarly to the City Section model seen before. This model was first introduced in [30] and is also implemented in the BonnMotion framework [1]. With respect to the City Section, the Manhattan model employs a probabilistic approach in the selection of nodes movements, since, at each intersection, a vehicle chooses to keep moving in the same direction with probability 0.5 and to turn left or right with probability 0.25 in each case. In the IMPORTANT framework implementation, the speed management is the same observed in the Freeway model, which adds realism with respect to the City Section model, where all cars traveling on the same lane have instead an identical velocity.

Note that the randomness and safety conditions with respect to the ahead driver imposed to nodes speed by the IMPORTANT implementation of the Freeway and Manhattan descriptions make these model fall somehow in between random mobility and car following theory. However, we classify the Freeway and Manhattan models as random mobility models because the car interaction rules they employ are too simple and do not reproduce a realistic drivers behavior.

Saha and Johnson [8] modeled vehicular traffic with a random mobility of nodes over real road topologies extracted from the maps of the US Census Bureau TIGER database [18]. In Sahas work, nodes select one point over the graph as their destination and compute the shortest path to get there. The edges sequence is obtained weighting the cost of traveling on each road on its speed limit (which is recorded by the TIGER format) and on the number of vehicles already moving on it, in a way to reproduce the real world tendency of drivers to avoid congested paths. The speed of a node is set to a constant value in the range $[v_{max} - \epsilon,\ v_{max} + \epsilon]$, where v_{max} is the speed limit of the road the car is moving on. All roads are considered bi-directional and single lane, and no car-to-car interaction is modeled.

Zhou *et al.* [12] propose a Real Track mobility model, derived from the Virtual Track model [10]. The Virtual Track model binds nodes movement over a graph, whose vertices are referred to as switch stations, and whose edges are defined virtual tracks. The edges have a length, equal to the distance between the switch stations they connect, and a width, which is user-defined. Thus, they can be graphically represented as rectangles more than lines. Nodes moves in groups, according to the Reference Point Group Mobility (RPGM) model [66], which defines a common direction for the group, and then adds some bounded randomness to the movement of the single nodes within the group with respect to the common direction. Groups of nodes are allowed to move following a Random Waypoint model [21] from one switch station to another switch station only within the virtual tracks, and only in the direction of the next switch station (i.e. the Random Waypoint is biased so that the next destination must be nearer than the current position to the target switch station). At switch stations, nodes may leave their current group and join other groups. The Real Track model applies the Virtual Track model to real road topologies, extracted, as it was the case for Saha, from the TIGER database. Road intersections are mapped into switch stations and roads into virtual tracks, to which a fixed width is assigned. The idea at the base of the Real Track model is to reproduce the clustering of vehicles occurring at intersections and propagating over the roads. However, the strategy chosen is not based on any vehicular traffic theory result, nor it is validated against other models or real world data. Moreover, since this model bases vehicles movements on the RPGM mobility model, the patterns are random and no car to car interaction is considered.

1.4.3 Traffic stream models

Traffic stream models look at vehicular mobility as a hydrodynamic phenomenon, and try to relate the three fundamental variables of velocity $v(x, t)$, density $\rho(x, t)$, and flow $q(x, t)$. All of them are functions of space x and time t, averaged over sufficiently large regions. Since they regard vehicular traffic as a flow, these models fall into the category of macroscopic descriptions. The basic equation comes from the idea that, given a road section, the number of vehicles can vary only due to cars entering or leaving the section. This leads to the following continuity equation:

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho v}{\partial x} \tag{1.1}$$

that is, the density over a small section dx varies, during a small interval dt, according to the corresponding flow. The simplest model of this kind was proposed by Lighthill and Whitham [55], assuming the velocity to be a function of the density:

$$\frac{\partial \rho}{\partial t} + \frac{d}{d\rho}(\rho v(\rho))\frac{\partial \rho}{\partial x} \tag{1.2}$$

which is capable of modeling kinematic waves. This model has been used along the last decades, but much more complex formulations, always based on the same assumptions, can be found in traffic flow literature.

Traffic stream models can handle large quantities of vehicles, at the cost of precision. This makes them interesting for high-level analytical studies of traffic behavior. However, they cannot model each vehicle independently, However, the study of VANETs is largely based on simulations which can rely on fast hardware, overcoming the problem of complexity. Moreover, in VANETs research the mobility of each vehicle is fundamental, as it dramatically impacts on key factors such as connectivity and links duration. As a consequence macroscopic level models are rarely found in communication networking literature.

A macroscopic approach is employed by Rudack *et al.* [50], again for a highway scenario. In fact, after presenting the fundamentals of traffic flow theory, the authors simply exploit common assumptions from the traffic flow literature, considering the distribution of vehicular speed as normal [60] and the distribution of vehicles inter arrival times at the beginning of the considered road section as exponential [47]. Since the goal of the paper is to determine the duration of connections between traveling vehicles and fixed gateways along the highway, no higher detail than the shape of these distributions is needed by the authors.

1.4.4 Car following models

In car following models the behavior of each driver is described in relation to the vehicle ahead, and, as they regard to each single car as an independent entity, they fall into the category of microscopic level descriptions. The first car following models date back 50s and, since then, they represent one of the most common ways to analytically represent vehicular traffic dynamics.

Most car following models compute the speed or acceleration of a car as a function of factors such as the distance to the front car and the current speed of both vehicles. A general delayed differential equation formulation, for a vehicle i, would thus be

$$\frac{dv_i(t)}{dt} = f(v_i(t), v_{i+1}(t), x_{i+1}(t) - x_i(t)) \tag{1.3}$$

where a common expression for f , named GHR after Gazis, Herman and Rothery, who first proposed it [22], is

$$\frac{dv_i(t)}{dt} = \alpha v_i^m(t) \left. \frac{v_{i+1}(t) - v_i(t)}{(x_{i+1}(t) - x_i(t))^l} \right|_{t-\tau} \tag{1.4}$$

where α, m and l are free parameter which can adapted to the traffic scenario that must be modeled. Note that the computation of the derivative at time $t - \tau$ accounts for the finite reaction time τ of drivers. Many other factors, characterizing both cars technical constraints and drivers attitude, can be detailed in car following models, increasing their level of realism. Also, car following models often include lane changing rules to regulate movement of vehicles between lanes. Car following descriptions integrated by lane changing rules can model the behavior of vehicular traffic on individual multi-lane roads, and are thus typically employed in highway mobility analysis. In simulation, they can also be used to model vehicular dynamics on independent roads of an urban scenario. In that case however, a proper accounting for interactions between traffic flows at road junctions must be provided. That is, rules for intersections crossing, in presence of stop/priority signs and traffic lights, must be defined as well. The same is hard to accomplish within analytical frameworks, as the joint complexity of the different acceleration, lane changing and intersection management descriptions leads to intractable problems [20].

Compared to macroscopic descriptions, these models are usually more precise but computationally more expensive, as their execution cost increases with the number of simulated vehicles. For this reason, it is to be remarked that differential formulations of the car following problem, such as GHR, are rarely used when large scale simulations have to be run. Instead, discrete time models, as those listed in the remainder of this Section, are employed. For a survey and comparison of car following models, along with discussion on their implementation in traffic simulators, the reader can refer to in [59, 49, 68].

In her thesis, Breisermeister [17] uses a microscopic level description for vehicular movements on a single-lane, bi-directional straight road. The car-following model is taken from Krauß [58, 44]. The model takes four input variables (the maximum velocity v_{max} , the maximum acceleration a, the maximum deceleration b and the noise η that introduces stochastic behavior to the model), discretizes the time with step Δt, and is built up by the following set of equations:

$$v_i^{safe}(t + \Delta t) = v_{i+1}(t) + \frac{\Delta x_i(t) - v_{i+1}(t)\tau}{\frac{v_i(t)+v_{i+1}(t)}{2b} + \tau} \tag{1.5}$$

$$v_i^{desired}(t + \Delta t) = min[v_{max}, v_i(t) + a\Delta t, v_i^{safe}(t + \Delta t)] \tag{1.6}$$

$$v_i(t + \Delta t) = max[0, v_i^{desired}(t + \Delta t) - \epsilon a\Delta t\eta] \tag{1.7}$$

The first equation computes the speed of vehicle i required to maintain a safety distance from its leading vehicle. The reaction time of the driver is represented by the time τ. The second equation determines the desired new speed of vehicle i, which is equal to the current speed plus the increment determined by the uniform acceleration, with upper bounds represented by the maximum and safe speeds. The third equation finally determines the speed of the following vehicle, by adding some randomness, in the measure of a maximum percentage η of the highest achievable speed increment $a\Delta t$ (η is a random variable uniformly distributed in [0, 1]). Also, note that in [17], only single-lane roads are simulated, thus no overtaking model is considered. However, in [44] a set of lane changing rules extending the above car following equations is proposed. Such rules determine a drivers willingness to change lane, as well as the safety requirements that must be satisfied for the lane change to occur. In particular, the lane changing model distinguish between two possible situations, namely a movement to the left lane and a movement to the right lane. Different rules are proposed in order to mimic the use of left lanes for over-takings and right lanes for free flow travel. In the following, we use the notation $|\cdot|_L$ and $|\cdot|_R$ to indicate the value of the expression within bars, computed as if the vehicle was in the lane at its left and at its right, respectively. First, a traffic congestion ($\mathcal{C}$) condition is introduced as

- IF $v_i^{safe}(t) < v_t^{safe}$ AND $|v_i^{safe}(t)|_L < v_t^{safe}$ THEN $\mathcal{C}$

where v_t^{safe} e is a threshold safe velocity, under which a lane is considered congested. Also, conditions for safe lane changes to left ($\mathcal{S}_L$) and right ($\mathcal{S}_R$) lanes are formulated as follows

- IF, FOR EACH i, $\Big(|v_i(t)|_X - b\Delta t \leq |v_i^{safe}(t)|_X\Big)$
  THEN $\mathcal{S}_X$ WITH X $\in$ [L, R]

where the inequality guarantees that, is the lane change is performed, each vehicle in the system is still able to brake with a finite deceleration b and avoid a collision with the vehicle ahead. Then, a lane change is considered favorable, with respect to a left ($\mathcal{F}_L$) and to a right ($\mathcal{F}_R$) movement, under the following conditions

- IF $v_i^{safe}(t) < v_{max}$ AND NOT $\mathcal{C}$ THEN $\mathcal{F}_L$
- IF $v_i^{safe}(t) \geq v_{max}$ AND $|v_i^{safe}(t)|_R \geq v_{max}$ THEN $\mathcal{F}_R$

meaning that a lane change to the left can be operated only if the current lane does not allow to reach the desired speed and if the traffic is not congested, as in that case changing lane would not increase the traveling speed. On the other hand, a movement to the right is allowed if the desired speed can be reached on both lanes, thus making occupancy of the left lane unnecessary. The lane change algorithm is then

- IF ($\mathcal{F}_X$ OR $\eta < p_{change}$) AND $\mathcal{S}_X$
  THEN MOVE TO X, WITH X $\in$ [L, R]

which simply states that the movement can be performed if the change is favorable and the safety condition holds. Some randomness is added to the process, by allowing the vehicle to change its lane even if not favorable, with a small probability pchange , as η is a random variable uniformly distributed in [0, 1]. Finally, the model prevents over-takings on right lanes, with the exception of those occurring in congestion regime, by means of the following condition

- IF $v_i(t) > |v_i^{safe}(t)|_L$ AND NOT $\mathcal{C}$ THEN $v_i(t) = |v_i^{safe}(t)|_L$

so that a vehicle in non congested traffic situation lower bounds its speed not only on the safety speed evaluated with respect to the front vehicle on the lane it is currently traveling on, but also on that computed with respect to the front vehicle on its left lane.

The same model by Krauß is employed within the Simulation of Urban MObility (SUMO) project [23], which is developing an open source traffic simulation package. In that case, both acceleration and lane changing rules are adopted. The tool also uses a dynamic user equilibrium approach by Gawron [35] to model realistic traffic assignments, and an original intersection management scheme to handle vehicle movement through road junctions. This tool is of interest for the VANET research community mainly because Karnadi *et al.* [32] developed the Mobility Model Generator for Vehicular Networks (MOVE) tool, which stands over SUMO and includes GUIs to facilitate the process of road topology and vehicular mobility definition, as well as output traces converters for network simulators as ns-2 and QualNet.

Haerri *et.al* [38] propose a vehicular mobility simulator for VANETs, called VanetMobiSim, which employs a different car following model from Treiber *et al.* [52], defined Intelligent Driver Model (IDM). Such model characterizes drivers behavior through the instantaneous acceleration of vehicles, computed through the following equations

$$\frac{dv_i(t)}{dt} = a\left[1 - \left(\frac{v_i(t)}{v_{max}}\right)^4 - \left(\frac{\delta}{\Delta x_i(t)}\right)^2\right] \tag{1.8}$$

$$\delta = \Delta x_{min} + \left[v_i(t)T + \frac{v_i(t)(v_{i+1}(t) - v_i(t))}{2\sqrt{ab}}\right] \tag{1.9}$$

In the first equation, δ is the so called desired dynamical distance, computed, as shown in the second equation, as a function of the minimum bumper-to-bumper distance Δx_{min} , the minimum safe time headway T, the speed difference with respect to front vehicle, and the maximum acceleration and deceleration a and b. When combined, these formulae give the instantaneous acceleration of the car, divided into a desired acceleration $1 - \left(\frac{v_i(t)}{v_{max}}\right)^4$ on a free road, and a braking deceleration induced by the preceding vehicle $\left(\frac{\delta}{\Delta x_i(t)}\right)^2$. A lane changing model from the same authors [51] is used to model over-takings in VanetMobiSim. The model, called MOBIL, follows a game theoretical approach, and allows a vehicle to move to an adjacent lane if his/her advantage, in terms of acceleration, is greater than the disadvantage of the back car in the new lane. The model also consider a politeness factor and a right lane advantage bias, which reproduce the real world behavior of drivers to stay on rightmost lanes. Adopting again the convention that $|\cdot|_L$ and $|\cdot|_R$ indicate the value of the expression within bars, computed as if the vehicle was in the lane at its left and at its right, and the concept of safe and favorable movement to the left ($\mathcal{S}_L$, $\mathcal{F}_L$) and to a right ($\mathcal{S}_R$, $\mathcal{F}_R$), MOBIL allows a car to change its lane under the following safety condition

1. IF $|\frac{dv_{j-1}(t)}{dt}|_X > -b_{safe}$ THEN $\mathcal{S}_X$, WITH X $\in$ [L, R]

that is, is the back vehicle in the new lane does not have to brake more than a safe deceleration value b_{safe} after the completion of the maneuver. Lane change are considered favorable, as follows

1. IF $|\frac{dv_i(t)}{dt}|_X - \frac{dv_i(t)}{dt} \pm a_{bias} \geq p\left(\frac{dv_{j-1}(t)}{dt} - |\frac{dv_{j-1}(t)}{dt}|_X\right) + a_t$ AND $\mathcal{S}_X$ THEN $\mathcal{F}_X$, WITH X $\in$ [L, R]

where the left hand side of the inequality is the advantage that the lane change would bring to the car under study and the right hand side represents the disadvantage brought by the same movement to the back car in the new lane. Thus, the lane change is favorable only if the advantage of the current driver is higher than the disadvantage of the new back driver. The a_{bias} acceleration term is subtracted when the movement is to the left, and added when the movement is to the right, so that vehicles tend to stay on their right. The p value models the drivers politeness, by weighting the disadvantage of the other drivers, while the a_t acceleration threshold avoids that lane hopping phenomena occur in borderline conditions. VanetMobiSim also adds to the acceleration and lane changing models intersection management techniques in presence of stop signs and traffic lights, so to reproduce vehicular mobility in urban environments.

The same model from Treiber is used by Jaap *et al.* [57], but without considering lane changes and with a simplified intersection management with respect to the previous work by Haerri.

Recent works by Gorgorin *et al.* [19] and Bononi *et al.* [45] employ implementations of a particular category of car following models, which is referred to as psycho-physical models [65, 60]. These models divide the bi-dimensional space of distance Δx and speed difference Δv with respect to front vehicle into several areas corresponding to different driver behaviors.

In an undergoing work to simulate a complete urban mobility, Kim *et al.* [39] use a desired-spacing model. Desired-spacing models perform car following between simulated vehicles by maintaining among them a given space headway [49]. In [39], a simple desired-spacing model is employed, which computes the headway of a vehicle i as a linear function of the current speed:

$$\Delta x_i(t) = \alpha + \beta v_i(t) \tag{1.10}$$

where α and β are two constants whose value is determined from real world observations [36]. The same framework implements a lane changing model, from [11], according to which the desire of a driver to change lane is modeled as a probability p_{change}:

$$p_{change} = \frac{1}{e^{\alpha M + \beta M(\bar{v} - |\bar{v}|_X)}} \tag{1.11}$$

where αM and βM are merging constants from real world measurements, while $\bar{v}$ and $|\bar{v}|_X$ are the average speeds of vehicles between the current node and the next intersection, on the current lane and on the lane considered for change (X $\in$ [L, R], according to the notation adopted for the lane change models already presented), respectively.

In [46], a mobility model based on cellular automata [42, 43] is used.

Cellular automata vehicular mobility models discretize the road in cells, each containing at most one vehicle. Then, some rules are applied to determine the speed of each car, according to nearby cells status, which depends on whether a vehicle is present in the cell and, if so, on its current speed. Since the level of details of this category of models reaches the behavior of the single vehicle, cellular automata descriptions belong to the microscopic class. Also, it is the front vehicle that constrains the speed of a driver, thus these models are also part of the car following descriptions.

In [46], one bidirectional highway is considered, with two lanes in each directions. In the following, we define as i the cell occupied by vehicle i. Nagels model determines the speed of i, according to the following algorithm:

- IF $v_i(t) < v_{max}$ THEN $v_i(t + \Delta t) = v_i(t) + 1$;
- IF $v_i(t + \Delta t) \geq \langle i + 1 - \rangle i$ THEN $v_i(t + \Delta t) = \langle i + 1 - \rangle i - 1$;
- IF $v_i(t + \Delta t) > 1$
  THEN WITH PROBABILITY p DO $v_i(t + \Delta t) = v_i(t + \Delta t) - 1$;

where the first line accounts for acceleration, the second for braking induced by slower cars ahead, the third introduces randomness in the process. Once the speed has been updated, the vehicle movement is performed, by changing its cell to $\langle i \rangle + v_i(t+\Delta t)$. This model also includes lane changing rules, which allow a vehicle to move to its left lane under the condition

- $v_{i+1}(t) \leq v_i(t)$ OR $v_{j+1}(t) \leq v_i(t)$;

that is, the lane change is granted if there is a slower front vehicle on the current lane, or on the left lane ($j + 1$ refers to the front vehicle on the left lane in this case). The second condition prevents that overtakings on the right occur, by forcing the car to move left and to try an overtaking using a lane further on the left. A movement from the left to the right is permitted under the condition

- $v_{i+1}(t) > v_i(t)$ AND $v_{j+1}(t) > v_i(t) + \Delta$;

where $j + 1$ refers to the front vehicle on the right lane. Thus, a car moves to its right if the ahead vehicles on the current lane and on the right lane are faster than it is. Δ is a so-called slack parameter, modeling the drivers tendency to stay on the left lane after overtaking another car. In both cases, the following safety conditions must hold

- $\langle j + 1\rangle - \langle i\rangle \geq v_i(t)$;
- $\langle i\rangle - \langle j - 1\rangle \geq v_{max}$;

meaning that there must be enough space on the new lane between the back car $j-1$ and the front car $j+1$ for the movement to take place. In particular, the front gap must be at least the velocity of i, while the back gap must match the maximum allowed speed.

Other related works in VANETs literature employing generic car following models are those from Choffnes *et al.* [24] and Mahajan *et al.* [9]. In both cases, basic car following techniques are employed jointly with intersection management mechanisms, which reproduce stop sign and traffic light behaviors at intersections.

1.4.5 Queue models

Recent work from Baumann *et al.* [14, 64] present very large scale (i.e. on a whole county) VANETs performance evaluations, using a vehicular traffic simulator from ETH [56]. In such tool, queue models are employed.

Queue models for were first introduced in the vehicular traffic field by Gawron [35]. According to the queue paradigm, each road is modeled as a FIFO queue, and each vehicle as a queue client. Each road queue k is characterized by its length l^k and a maximum flow q^k_{max}, determined by the number of lanes. Every time a vehicle enters a road, a travel time is computed, depending on the desired free flow speed of the driver v_{max}, on the number of vehicles on the road n^k and the road length. It has been shown in [35] that even a simple expression of travel time $\frac{l^k}{v_{max}}$, which neglects the effect of vehicular density on the speed, leads to very good approximations of results obtained with much more complex microscopic mobility models.

The car is then enqueued in the priority queue of the road, according to the travel time calculated before. At every time step, vehicles whose travel time has expired can be removed from the head of the queue and inserted into the queue representing the next road in their trip. However, when multiple choices are available to exit a road, an intermediate step in necessary, and first-in-first-out queues are added for each outgoing flow. In that case, vehicles at the head of the priority queue are moved to one of the FIFO queues, depending on their destination. The FIFO queues have a finite capacity, meaning that only a certain number of vehicles per second can access them. Since the movement from one road to another is constrained by the capacity of such next road, which can be, a vehicle at the head of an output queue can join the following queue only if there is space on the following road. The capacity of a road is easily modeled as $c^k = \frac{n^k q^k_{max}}{x_{min}}$, where x_{min} is the distance between the front of two adjacent vehicles in jam conditions. Thus, if the new road has c^k cars already queued, it will not accept further vehicles, and drivers willing to enter the road will have to wait until a spot is freed. Note that more complex expressions for the travel time and capacity can be easily implemented, and realistic intersection management can be modeled [56], e.g. giving different priorities to multiple queues accessing the same road.

Since queue models describe the movement of each vehicle in an independent way, but also with a minimal level of detail, they fall into an intermediate category with respect to macroscopic and microscopic descriptions, which can be referred to as mesoscopic. Queues models have very low computational cost, because they update the status of a vehicle only when a vehicle enters a new priority or FIFO queue. This allows to model very large road topologies, up to hundreds of thousands of vehicles. The drawback is the reduced realism of the outcome, which is less precise than that obtained with other models (e.g., queue models do not reproduce shock-waves caused by periodic perturbations, a common phenomenon in vehicular traffic).

1.4.6 Behavioural models

In [34, 33] Legendre *et al.* introduced a novel approach to the problem of modeling human mobility, which can be applied to vehicular traffic as well. The approach is called behavioural modeling and is borrowed from the fields of biological physics and artificial intelligence. The key idea is that every movement is determined by behavioural rules, which are imposed by social influences, rational decisions or actions following a stimulus-reaction process. These rules can be modeled as attractive or repulsive forces. In the case of vehicular mobility, the next intersection towards the trip destination wields an attractive force on the vehicle, whereas other vehicles or obstacles in general exert a repulsive force on it. The result from the composition of these forces determines the acceleration vector which drives the car movement. However, this model is especially expensive under the computational point of view, as every movement require the elaboration and composition of multiple inter-object forces.

1.5 Generation of mobility patterns

The simple Freeway model and Manhattan (or Grid) model were the initial steps, then more complex projects were started involving the generation of mobility patterns based on real road maps or monitoring of real vehicular movements in cities. However, in most of these models, only the macro-mobility of nodes was considered. Although car-to-car interactions are a fundamental factor to take into account when dealing with vehicular mobility, little or no attention was paid to micro-mobility.

Recently, new open-source tools became available for the generation of vehicular mobility patterns. Most of them are capable of producing traces for network simulators. In the rest of this section, we review some of these tools, in order to understand their strengths and weaknesses.

The IMPORTANT tool and the BonnMotion tool [31, 1] implement several random mobility models, plus the Manhattan model. While the IM-

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

The GEMM tool [54] is an extension to BonnMotions and improves its traffic generator by introducing the concepts of Attraction Points (AP), Activity and Role. Attraction points reflect a destination interest to multiple people. Activities are the process of moving to an attraction point, while roles characterize the mobility tendencies intrinsic to different classes of people. While the basic concept is interesting, its implementation in the tool is limited to a simple RWM between APs. It however represents an initial attempt to improve the realism of mobility models.

The MONARCH project [8] proposed a tool to extract road topologies from real road maps obtained from the TIGER database. The possibility of generating topologies from real maps is considered in the framework, however the complete lack of micro-mobility support makes it difficult to represent a complete mobility generator.

The Mobility Model Generator for Vehicular Networks (MOVE) was recently presented as an on-going work [32]. It seems a quite complete tool, featuring real map extrapolation from the TIGER database as well as pseudo-random and manual topology generation. No micro-mobility and complex traffic generation are considered yet, but the in-progress status of the project allows us to think that this might be corrected in the near future.

The Street Random Waypoint (STRAW) tool [24] is a mobility simulator based on the freely available Scalable Wireless Ad Hoc Network Simulator (SWANS). Under the point of view of vehicular mobility, it provides urban topologies extractions from the TIGER database, as well as micro-mobility support. STRAW is also one of the few mobility tools to implement a complex intersection management using traffic lights and traffic signs. Thanks to this, vehicles are showing a more realistic behavior when reaching intersection. STRAW contains accurate mobility constraints as well as a realistic traffic generator engine. STRAW also includes several implementations of transport, routing and media access protocols, since they are not present in the original SWANS software. The main drawback of the tool is the very limited diffusion of the SWANS platform.

The GrooveSim tool [29] is a mobility and communication simulator, which again uses files from the TIGER database to generate realistic topologies. Being a self-contained software, GrooveSim neither models vehicles micro-mobility, nor produces traces usable by network simulators. The interesting feature of this model is the non uniform distribution of vehicles speeds. Indeed, motion constraints such as speed limitations, often force vehicles to give up in their effort to reach the velocity initially set by the model. Although that is might look as a straightforward pattern, this type of motion constraints is, at this time, considered only by a few simulators. GrooveSim includes four types of velocity models, where the most interesting is the road-based velocity when used in conjunction with a shortest trip path generation. The authors illustrated how vehicles were naturally choosing the roads with the highest speed limitations while on their journey. The main drawback of this tool is however its lack of a micro-mobility model as well as mobility traces for network simulators.

The CanuMobiSim tool [2] is a tool for the generation of movement traces in a variety of conditions. Extrapolation of real topologies from detailed Geographical Data Files (GDF) are possible, many different mobility models are implemented, a GUI is provided, and the tool can generate mobility traces for ns-2 and GloMoSim. Unlike many other tools, the CanuMobiSim tool keeps micro-mobility in consideration, implementing several car-to-car interaction models such as the Fluid Traffic Model, which adjusts the speed given vehicles local density, or the Intelligent Driver Model (IDM), which adapts the velocity depending on movements between neighboring vehicles. Also unlike other tools, CanuMobiSim includes a complex traffic generator that can either implements basic source-destination paths using Dijkstra-like shortest path algorithms, or similarly to the GEMM, it can model trips between Attraction Points depending on the class of users specific motion patterns. This solution is actually the only fully implemented and available solution considering heterogeneous classes of user and destinations. In order to improve its modeling capability, CanuMobiSim has even been recently extended by the same authors and now includes radio propagation information for ns-2 and GloMoSim.

The City Model [57] has been basically designed for routing pro tocols testing and no network simulator traces are provided. Unlike GrooveSim, this model includes the IDM. However, this simulator falls short from realistically representing vehicular motions mostly due to the unique grid-based mobility con- straints it includes.

The SSM/TSM model [9] represents actually two different mobility models, a Stop Sign Model and a Traffic Sign Model. The motion constraints part is dealt using a TIGER parser, while the traffic generator includes the Car Following Model. As GrooveSim, both SSM and TSM include a road-dependent velocity distribution. However, this model goes farer than GrooveSim, since it contains a basic traffic generator which makes its mobility traces more realistic than GrooveSims. And similarly to STRAW, SSM/TSM has been specifically designed to model vehicles motions at intersections. The authors managed to show how a basic intersection management such as a simple stop sign was able to bring out a clustering effect at those intersection. In urban environment, this effect is better known under the name Traffic Jam, and is hardly represented in most of the actual simulators.

New solution named MobiREAL has been recently presented [63]. Although that it seems more focused on the modeling of pedestrian mobility, its strict compliance with the framework and its novel approach of cognitive modeling makes it very promising for a future extension to vehicular mobility.

The most interesting features is that MobiREAL enables to change a node or a class of nodes mobility behavior depending on a given application context. At this time, only CanuMobiSim, VanetMobisim and MobiREAL are able to include this feature. This particular application context is modeled by a Condition Probability Event (CPE), a probabilistic rule-based mobility model describing the behavior of mobile nodes, which is often used in cognitive modeling of human behavior. As most of recent mobility models, MobiREAL is able to include geographical information. Moreover, it is also able to use this information to generate obstacles and more specifically it is able to model radios interference and attenuations on the simulation field. With CanuMobiSims extension and the Obstacle model, they are the only models that are able to both generate motion traces and signal attenuation information. MobiREALs major drawback at this time is the limited diffusion of Georgia Tech Network Simulator (GTNets) and the manual configuration of all necessary parameters, which requires a full recompilation of the simulator at each reconfiguration.

1.6 About VANETs applications

Vehicle-to-vehicle communication is a very challenging topic in recent years. Vehicles equipped with devices capable of short-range wireless connectivity can form a particular mobile ad-hoc network, called a Vehicular Ad-hoc NETwork (VANET). The existence of such networks opens the way for a large range of applications. We consider that two of the most important classes of such applications are those related to route planning and traffic safety.

1.6.1 Traffic safety

Safety applications involve disseminating urgent information, which is unavailable in the drivers field of view, or is difficult to notice. Examples of this information are the presence of fog, obstacles obstructing the line of sight. Many accidents occur in foggy conditions, because the drivers notice too late that some kind of accident that occurred in front of them. Safety at intersections could also be enhanced, because the risk of collisions could be detected in advance and the driver could be warned seconds before the possibility of an imminent collision. In case of accident vehicles through the network could alert directly the hospital and the police.

1.6.2 Route planning

Traffic Congestion is a problem in Americas 437 urban areas and it is getting worse in regions of all sizes. Congestion caused urban Americans to travel 4.2 billion hours more and to purchase an extra 2.9 billion gallons of fuel for a congestion cost of $78 billion [67, 53, 40]. This was an increase of 220 million hours, 140 million gallons and $5 billion from 2004. Also, traffic congestion does not allow an efficient usage of the urban roadways. Often the shortest path is not the fastest one in metropolitan areas, therefore modern GPS navigators in addition to give wrong fastest path, increase traffic congestion because they direct vehicles in same roadways. One possible remedy to the congestion is the exploitation of the vehicle-to-vehicle communication.

Route planning aims to provide drivers with real-time traffic information, which would, in the absence of a VANET, require expensive infrastructure. By contrast, the VANET approach is highly scalable and has very low maintenance costs. Moreover, short-range wireless communication technologies (such as 802.11) have no associated cost, other than the communication device. Therefore vehicles in VANETs change path dynamically on the basis of fastest path.

1.6.3 Evaluating VANETs

Most applications to be deployed on top of a VANET require some sort of data-dissemination model. This is a challenging problem, due to the unique characteristics of a VANET. Such a network has a very high degree of nodes mobility and a very large scale. Network partitioning, which is the breakdown of a connected network topology into two or more separate unconnected topologies, occurs frequently, making end-to-end communication impossible sometime. Several studies [16] show that the performance of classical, topology-based routing protocols in vehicular networks is poor, due to the extremely high mobility of the nodes.

The evaluation of VANET protocols and applications could be made through real outdoor experiments, which should involve a large number of nodes, in order to obtain significant results. However, performing such large-scale experiments is extremely expensive in term of both money and time. Therefore, simulation is an indispensable tool.

1.6.4 Simulating VANETs

The simulation of a VANET requires two different components:

1. a network simulator, capable of simulating the behavior of a wireless network;
2. a vehicular traffic simulator, able to provide an accurate mobility model for the nodes of a VANET.

Simulating a vehicular network involves two different aspects. First, there are issues related to the network, such as medium access control, signal strength, propagation delays. The network simulators, like *The Network Simulator NS-2* [5] and *Jist/SWANS* [4], cope with such issues. However, a general-purpose wireless network simulator is by no means enough for an accurate simulation of a vehicular network. Nodes in a wireless network usually move according to the random-waypoint model. This means that they have an origin and a destination and move towards the destination. But vehicles only move along roads and that is a very particular situation. Furthermore, real vehicles move according to very particular traffic models, due to the street topology, intersections, traffic regulations and drivers behavior. Hence it is of key importance for a vehicular network simulator to have a mobility model as close as possible to real vehicular mobility. As recent studies [24] have proven the use of an inaccurate mobility model, like the popular random waypoint model (which may work for some mobile ad-hoc networks, but is definitely not an accurate representation of mobility in a VANET), can lead to erroneous results [24, 7].

Vehicular traffic simulators can be classified in macroscopic and microscopic simulators. Macroscopic simulators deal with global measures, like traffic flow, while microscopic simulators take into account the movement of each particular vehicle. Many commercial vehicular traffic simulators are available. They have not been designed especially for the simulation of the vehicular networks. They are primarily used to study the traffic, in order to validate projects, like building a new road, or a new tram line, or for designing effective traffic signals.

An example of a commercial vehicular traffic simulator is VISSIM [6] that is a microscopic simulator implementing driver behavior models, like car-following or lane changing. According to its producers, it is used in over 70 countries.

An integrated simulator was developed by a team at Northwestern University. It is based on an original vehicular traffic model, called Street Random Waypoint (STRAW). The authors have used the simulator in order to prove that studying routing protocols for a vehicular network without an accurate vehicular traffic model is a wrong approach. In this respect, they compared results obtained with the Random Waypoint model (which is a very inaccurate representation of a vehicular network) with the results obtained with the STRAW model. Their experiments clearly indicate that using the Random Waypoint model will not produce accurate results for a vehicular network.

However, we believe the mobility model implemented in existing simulators [24, 7, 29] is not a sufficiently accurate representation of real vehicle mobility. Thus, the simulator of Saha and Johnson uses real maps, in the TIGER format and vehicles move along the streets. Each vehicle moves completely independent of other vehicles, with a constant speed randomly chosen. Multi-lane roads or traffic control systems are not taken into consideration.

Other authors [29] make the same oversimplifying assumptions and do not consider multi-lane roads or car-following models.

The mobility model of [24] is more complex. It also uses TIGER files, and considers car-following models. The motion of a vehicle is influenced by the preceding vehicle. The authors also implement traffic control systems: timed traffic lights and stop signs. However, multi-lane roads are not taken into consideration.

Furthermore, the majority of VANET applications implies that vehicles react to messages. For instance, if a driver receives a message saying that the road ahead is congested, that driver will change its route. In order to study such reactions, combining an existing vehicular traffic simulator with an existing wireless network simulator is not possible. An integrated simulator is needed.

Based on these aspects, we have chosen to develop a VANET simulation tool, integrating vehicular mobility and wireless transmission simulator.

1.7 Goals of this work

In the proposed work we addressed two goals:

1. we developed an integrated simulator, that can mix vehicular traffic simulator and selected wireless network simulator characteristics;
2. we investigate the behaviour of some basic communication protocols to avoid traffic congestion.

Chapter 2

The VANET Simulator

The evaluation of VANET protocols and applications could be made through real outdoor experiments, which should involve a large number of nodes, in order to obtain significant results. However, performing such large-scale experiments is extremely difficult. Therefore, simulation is an indispensable tool.

We have developed a simulation tool, comprising a microscopic traffic simulator and a wireless communication model.

The simulator has been developed for urban traffic simulation. The extension to longer distance traffic does not present any conceptual difficulty.

2.1 An architectural overview

The architecture of the simulator is shown in Figure 2.1. The VANET simulator receives two inputs: a *map* of the area which will be simulated, and a *traffic scenario* which contains the vehicles that will be loaded into the simulator.

It is composed by three modules: the *traffic simulator* subsystem, the *network simulator* subsystem, and the *time* module that models the time inside the simulator. These time ticks are used by all entities inside the simulator.

Finally, the simulator produces the *simulation results* as output.

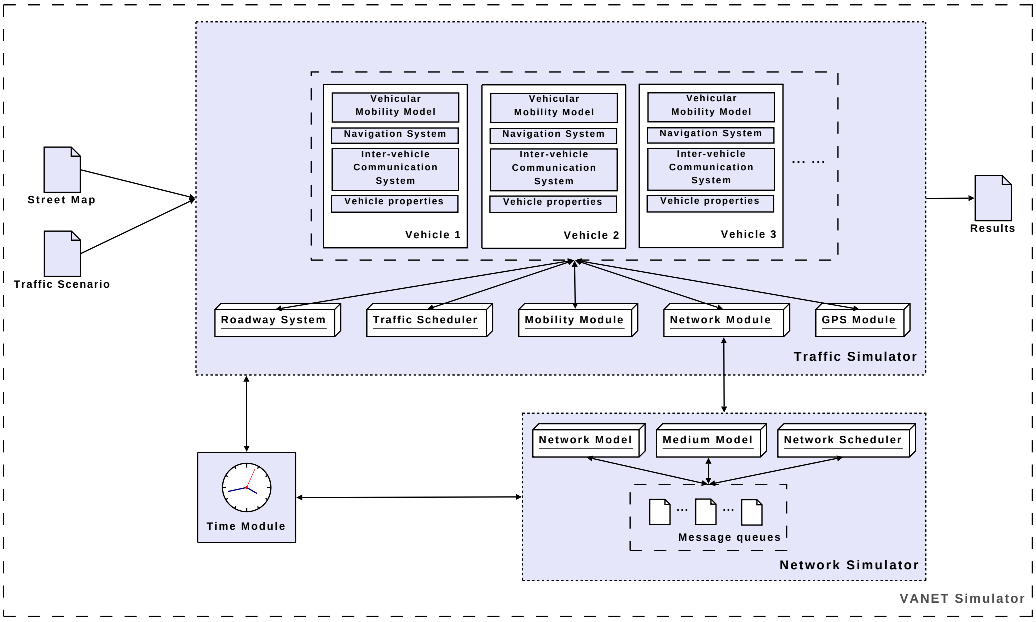

Figure 2.1: Simulator Architecture

2.1.1 Time module

The simulation time is modelled by the *time module*. It is implemented through the Singleton [28] design pattern, therefore there is only one instance of it inside the simulator and every entity of the simulator using the simulation time is synchronized on it.

The simulation time advances with a fixed time resolution (*tick*), which can be set manually with a command line parameter, after executing the application code for the current simulation time. The default fixed time resolution is 1 s.

2.1.2 Street map

A digital map is required in any kind of VANET application. Each vehicle which is part of the system should have such a digital map. Also the digital map is used by *Roadway System* module to build the simulated roadway system.

In our simulator, we have chosen to use XML files because we want simulate the roadway system of Pisa. Other formats can be simply loaded into the simulator extending the default loader.

The XML files contain detailed geographical information about all the roads in a region, from large highways to small streets. The contained data comes in the form of geographical coordinates (latitude, longitude) for the roads. Thus, for every road, the files specify its end point, together with as many intermediate points that are needed to reproduce the shape of the road.

A road is described by the following properties:

1. a unique identifier is used to refer the road inside the simulator;
2. the start point identifies the geographical coordinates of the intersection where the road starts;
3. the end point identifies the geographical coordinates of the intersection where the road ends;
4. the length is the distance between the start point and the end one;
5. the number and the width of lanes;
6. the speed limit corresponding to the road since a street can have a lower limit than an highway.

2.1.3 Traffic scenario

The traffic scenario is an XML file that contains the vehicles that will be loaded into the simulator. It can be generated using a utility bundled with the simulator. Every row of the file specifies the type of vehicle and its properties, as follow:

1. an XML tag specifies the type of vehicle (for example *car* or *truck*);
2. the attributes *source* and *destination* set start point and the destination;

3. the attributes *length, width, speed, acceleration* and *deceleration* set characteristics of the vehicle;
4. the attributes *navigator* set the navigation system of the vehicle, for example *gps* is normal GPS navigator (using Dijkstra's shortest path algorithm), *igps* is the navigator which implements the centralized protocol (see Section 3.2) and *cgps* is the one which implements the collaborative protocol (see Section 3.4).
5. the attribute *loop* specifies if the vehicle is loaded again when it reaches the destination;
6. the attribute *monitored* specifies if the simulation results of the vehicle will be included in the output;
7. the attribute *quantity* sets the number of vehicles that will be loaded.

2.1.4 The traffic simulator

The traffic simulator is the module that the builds simulated area from the street map and then simulates the traffic behaviours on it. The main entities that form the traffic simulator are:

1. the *roadway system*, which models the simulated area;
2. the *traffic scheduler*, which moves the vehicles on the roadway every system tick;
3. the *mobility module*, which describes the driver behavior model;
4. the *network module*, which models the wireless network used by the communication systems that equip the vehicles;
5. the *GPS module*, which models the GPS system used by the navigation systems that equip the vehicles;
6. the *vehicles*, which models the traffic status on the roadway system.

2.1.5 The network simulator

The network simulator, which is used by network module of the traffic simulator, copes with the delivery of messages from one node to another. It offers a set of network primitives that can be called by the communication system of vehicles of the simulator. Of special interest are the MAC and physical layers that determine VANET applications performance (Takai *et.al.* [61]). Since the focus of the proposed work is the traffic simulator and the mobility model, then the MAC and physical layers are been implemented following the simplifying assumptions described in Chapter 3.

2.2 The roadway system module

The digital map is converted in a directed graph by the roadway system module as follow:

1. an edge, models a road of the digital map.
2. a vertex models an intersection between two or more road.

2.2.1 Roads

For each road the starting and the ending intersection is specified. The roads are assumed as straight line between the two connected intersections. From the geographical information of the two extremes of the road it is possible to associated a geographical information to each point in the road.

The assumption of the road being a straight line seems too simplistic. However by adding other intermediate nodes where the curvature of the road is high, it is possible to adapt the simulated shape to the real shape.

A road can be one way or not. In both case each direction has one or more lanes. Each lane has a number which identifies it. Lane with the smallest identifier is used first until it is full, then are used the others following the order of identifiers.

Each lane of the road is managed as a First In First Out (FIFO) queue as follow:

1. the first vehicle in the lane is the first vehicle in the queue;
2. the last vehicle in the lane is the last vehicle in the queue;
3. the queue has a capacity C:
    (a) $C = L_r$ when there are not vehicles on the lane. L_r is the length of the road;
    (b) $C = 0$ when the last vehicle is at the beginning of the lane;
    (c) $C = d$ in other cases. Value d is the distance between the end of the last vehicle and the beginning of the road.

A road is full when $C = 0$ for all its lanes.

2.2.2 Intersections

Each intersection maintains information about all the incoming and outgoing roads. Also a geographical information of the intersection is maintained.

At each intersection there must be at least two roads, but at the boundary of the map there are special intersections, where only one road is specified.

An intersection can be managed by a traffic light, stop signals or traffic circle. The type of managing sets out the delay and the order that characterize the exit from the incoming roads and the enter in the outgoing ones. Other types of managing are possible simply extending the implemented classes.

2.3 Traffic Scheduler

The traffic scheduler is the entity which moves the vehicles on the roadway system. It follows the rules below:

1. it requests a list of intersections to the roadway system module;
2. for each intersections in that list, it gets the list of roads;
3. for each roads, it get the list of incoming lanes;

4. for each lanes, it computes:
    (a) the new position of the first vehicle according to the type of intersection manager, for example if intersection is managed by a traffic light the first vehicle is stopped when the light is red, it decelerates when the light is yellow and it is moved to the new entered road when the light is green;
    (b) the new position of the other vehicles;
5. when the new position of all vehicles is computed, it updates the old positions to the new ones.

2.4 The mobility module

The mobility module is the entity which models the driver behaviour inside the simulator. It is used by the traffic scheduler to move the vehicles.

Next, we briefly describe the driver behavior model we have implemented, which is based on the idea developed by Wiedemann [65], and further studied by Fellendorf and Vortisch [48].

The basic assumption is that a driver can be in one of four modes: free driving, approaching, following or braking.

Free driving means there is no influence from preceding vehicles on the same lane. In this situation, the driver will seek to obtain and maintain a desired speed. The desired speed and the acceleration depend on the driver personality, and on the road characteristics.

In the *approaching* mode there is a slower, preceding vehicle that influences the driver. In this situation, she/he will apply a deceleration in order to obtain the same speed as the preceding vehicle. The deceleration is a function of the distance between the two vehicles, their speeds, as well as other parameters.

The *following* mode means there is a preceding vehicle, but the speeds of the two vehicles are practically equal. In this situation, the driver will seek to keep the speed constant.

The *braking* mode means there is a slower preceding vehicle, very close in front. In this mode, due to the immediate danger, the driver will apply high deceleration rates.

We have also implemented a lane-changing model, for multilane roads. The model we have implemented is based on the lane-usage rules valid throughout most part of Europe. Thus, the usage of the first lane is required, unless it is occupied. It means that a driver will always try to stay on the lower lanes, except when overtaking another slower vehicle. Overtaking on the right side is not allowed. These rules are not valid in city environments, near intersections, where lanes are selected based on the direction the driver intends to follow.

The lane-changing model we have designed and implemented is based on a hierarchy between the four driving modes. Whenever a driver is in a different mode than free driving, she/he will always check if the higher lane can provide a superior mode. If that is the case, the driver will switch to a higher lane. Similarly, whenever a driver is in a different mode than braking, she/he will always check if the lower lane provides at least similar conditions. If that is the case, the driver will switch to a lower lane. The order of these checks is important. The higher lane is checked first. Thus, if a driver uses lane 2 and approaches another slower vehicle, it will first check if lane 3 is empty, and if that is the case it will switch to lane 3 (only if it can safely complete the switch, without interfering with any vehicles approaching from behind). If it had first checked the lower lane, it could have discovered that it is empty and it would have decided to use lane 1 for overtaking the vehicle on lane 2, which is forbidden in most European countries. However, it is not forbidden in the United States, where any lane can be used for overtaking. US traffic could easily be simulated, by making a random decision whether to first check the higher lane or the upper lane when looking for superior driving conditions.

We have also incorporated traffic control systems into our driver behavior model implementation. Thus, the vehicles we simulate are aware of traffic lights, priority roads and yield or stop signs, and their motion is simulated according to these traffic control systems.

Different driver profiles (aggressive, regular, calm) can easily be modeled by using the numerous model parameters. Each driver class is represented by a certain set of values for the parameters. In order to further differentiate the drivers, there is also a small deviation from the specified values, deviation computed randomly for each driver.

2.5 Vehicles

Vehicle are the objects that we simulating. Each vehicle has the following features:

1. an *origin*, which is specified in the XML file called *Traffic Scenario*. It is represented by a road identifier and a position along the road, where the vehicle is placed at the beginning. If the position along the road is left unspecified, then it will be used the beginning of the road;
2. a *destination* is modelled as the origin at it determines the final destination of the vehicle. If it is left unspecified, it is assumed equal to the beginning of the road;
3. a *time of departure*, which is the instant when the car appear at the origin;
4. a flag *loop* which specifies the behaviour of vehicle when it reaches the destination:
    (a) if true, the vehicle is started again at the origin;
    (b) if false, the vehicle is removed from the roadway system;
5. a flag *monitored* which specifies whether we must save the information about the travel, as the average speed, the travelling time, . . . ;
6. some vehicle's characteristic as: the *length*, the *maximum speed*, the *maximum acceleration*, the *maximum deceleration*;
7. a *vehicular mobility model* which is based on the vehicle's characteristic and the mobility module of the traffic simulator;

8. a *navigation system* which provides the information about the vehicle's position. It uses the GPS module of the traffic simulator. Notice that from position of the vehicle along the lane and the geographical information of the two extremes of the lane, it is possible to determine exactly the geographical position of the vehicle in the lane;

9. a *inter-vehicle communication system* which provides the communication services using the network module. It makes possible the message exchange between vehicles.

Chapter 3

Communication Protocols

All the vehicles are equipped with a radio that allows receiving and sending messages. The communications between vehicles is ruled by dedicated protocols that will be described in this chapter. The purpose of the protocols is to make all the vehicles aware of the traffic conditions. We focus our attention on two protocols:

1. a *centralized protocol*, where the information regarding the traffic condition are collected and transmitted by a unique server;
2. a *collaborative protocol*, where the traffic information are spread around by the vehicles.

Both the protocols are implemented in this work, and extensively tested by simulation as we will show in Chapter 4.

3.1 Communication model

Since the goal of our work is to evaluate the effectiveness of the communication protocols, we performed several simplifications on some lower level details that we believe do not affect the result of the simulations. Below we report all the simplifications.

1. The radio on each vehicle is simply characterized by a transmission radius R. All the vehicles are characterized by the same transmis-

sion radius R. This implicitly means that antennas are *isotropic* (see Figure 1). Although it is well known that the transmission radius is non-isotropic [69], we believe that this assumption does not bias the performance of the communication protocols the we will be studying.

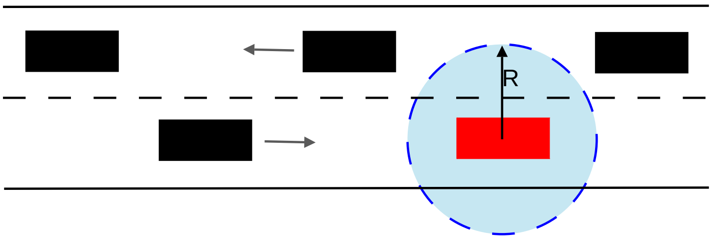

Figure 3.1: Transmission radius of communication system.

2. A connection can be established between two vehicles or a vehicle and base station antenna when their distance is smaller than R.
3. The delay for the establishment of the connection is zero.
4. Two vehicles within a distance R can communicate with each other with a zero delay (we neglect the collisions on the medium, the transmission delay, and the propagation delay).
5. The transmission medium is reliable (no message is lost, no message is corrupted).

All these simplifications are performed because our focus is not the study of the network model. Instead, we are focusing on the mobility model and the way the information can be diffused through the vehicles thanks to the communication protocols.

3.2 The Centralized Protocol

In the centralized protocol the vehicles are not allowed to communicate with each other. Instead, they send (and receive) information to a central server that stores and updates traffic information database (see Figure 3.2).

3.2.1 Communication Infrastructure

This protocol takes advantage of a centralized management of the information. This means that the communication system that needs a network infrastructure to connect and update traffic information from central server. Possible infrastructures that can enable this communication systems are:

1. mobile phone infrastructure (for example: GPRS or UMTS);
2. WiFi networks infrastructure;
3. sensor networks infrastructure.

The first solution uses an existent infrastructure, the mobile phone system, to connect vehicles to a central server. The usage of the mobile phone system has no startup cost to create infrastructure, because it already exists. However, it is expensive for the end users that pay connections to the Internet. The second and third solution are expensive, because the infrastructure needs to be deployed by the manager of the urban network.

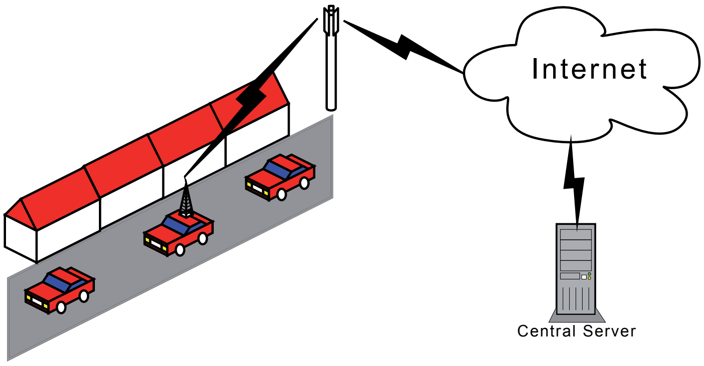

Figure 3.2: Scheme of communication in centralized approach

3.2.2 Implementation

In the centralized protocol two kind of entities are present: the server and the vehicles.

The server monitors the status of the traffic and provides this information to all the vehicles that require it. The stored information are

1. the map of the area where the vehicle are moving (for example a city map);

2. a table that stores the most updated information regarding the travelling times of each road on the map.

The format of the map can be the TIGER format [18] or something similar. It can be selected by the manufacturer of final product. The table that stores travelling times is a collection of rows whose format is shown in Table 3.1. The field id is an integer that identifies unequivocally a roadway inside the network. The field travellingTime contains the time needed to move from the start to the end of roadway identified by id. Finally, the field timestamp expresses *freshness* of the information about travelling time. In this case, a high value means a fresher information.

id	travellingTime	timestamp
32 bits	32 bits	32 bits

Table 3.1: Scheme of information about travelling time.

The algorithm of the centralized approach requires that each vehicle performs the actions listed below.

1. When the vehicle starts travelling, it requests to the central server an updated table of the travelling times.

2. The vehicle calculates the path from the start to the end of the roadway. It uses the Dijkstra shortest path algorithm [26] on the graph of the network, where the weights are set equal to the travelling times. Hence we can say that the it uses the *fastest-path algorithm.*

3. When the vehicle moves along a roadway, it monitors its own travelling travelling time.

4. When the distance to the end of the roadway is smaller than a threshold value D, the vehicle requests to the server an update of the travelling times table and, if necessary, it updates its fastest path.

5. When the vehicle reaches the end of the roadway it sends to the central server the information about the experienced travelling time of that roadway.

What value do we select for D? The value of D must balance two opposite needs:

- it should be small enough to have the freshest possible information;

- it should be large enough to allow the vehicle to compute the new fastest path before the next intersection.

An upper bound for D is given by:

$$v(T_{\mathsf{update}} + T_{\mathsf{dijkstra}}) < D \tag{3.1}$$

where v is speed of vehicle, T_{dijkstra} is the run time of Dijkstra's algorithm to find fastest path, whose complexity is $\mathcal{O}(K^2)$, where K is number of roadways, and T_{update} is the time required to download update of the travelling times and that can be expressed by:

$$T_{\mathsf{update}} = \frac{N_{\mathsf{table}}}{R_b} \tag{3.2}$$

$$N_{\mathsf{table}} = KN_{\mathsf{row}} \tag{3.3}$$

where R_b is bitrate of communication link, for example 54 Mbit/s (WiFi), 80 Kbit/s (GPRS), 177–236 Kbit/s (EDGE), 384 Kbit/s (UMTS), 7.2 Mbit/s (HSDPA), N_{table} is dimension of travelling times table, K is number of roadway in the map and N_{row} is dimension of single row. In the proposed work we used the map of Pisa which has approximately $K = 1500$ roadways, therefore $N_{\mathsf{table}} = 150$Kbit and $T_{\mathsf{update}} = 1$ sec if the communication link has $R_b = 150$ Kbit/sec.

Since a smaller value of D allows to have fresher information, we now address the problem of reducing the value of D.

Each time the Dijkstra's algorithm for the fastest path is run, the search for a new better solution can take advantage of the last solution. This allows to avoid the search from scratch and, in the end, to reduce the value T_{dijkstra}.

Another improvement is possible by reducing the number of roadways K whose travelling time need to be transmitted. For example we could select for transmission, only the travelling time of the roadways belonging to all the possible paths from the current position to the final destination. In our tests of the map of Pisa, we were able to decrease K by factor between 50 to 100.

3.3 The Collaborative Protocol

The collaborative protocol does not need a communication infrastructure for detecting and signaling the traffic conditions. In fact in this case the communication occurs only from vehicle to vehicle.

3.3.1 Communication without infrastructure

In a *Vehicle Ad-hoc NETwork* (VANET) the vehicles can be connected with each other through a short-range communication system.

The network topology in VANETs can be change frequently as effect of the network partitioning due to the high mobility of vehicles. Therefore it can happen that the communication is possible only between a small groups of vehicles, for example between those ones that move along the same street, and a vehicle can become unreachable at any time. An example of a dynamically changing topology is illustrated in Figure 3.3.

The communication between nodes can occur in two different ways, as shown in Figure 3.4:

1. *single-hop*, if the sender and the receiver are within the distance R;

2. *multi-hop*, if the sender and the receiver can exchange messages also

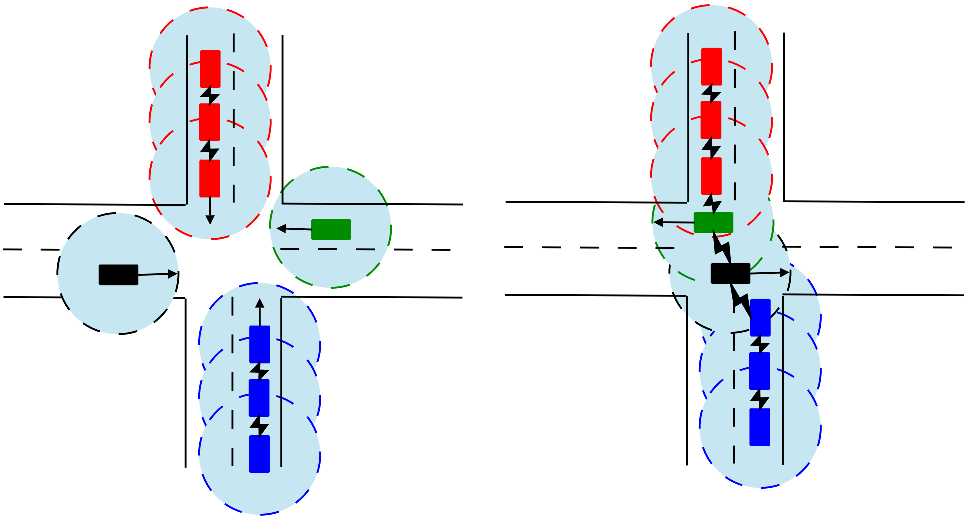

Figure 3.3: The network topology is dynamic in VANETs

when their distance is greater than R, by using routing the messages through intermediate nodes.

Hence, in the scenario of the collaborative protocol, the information is disseminated in the network. The effect of this information dissemination are that:

1. none of the vehicles can have a complete view of the network topology, therefore it is not possible to know the vehicles that can be reached;
2. the information reaches the receiver with unpredictable delay, since it highly depends on the possibility that a vehicle transports the information at the desired place.

These effects must be taken into consideration during the design of a VANET application.

3.3.2 Implementation

In our implementation of the collaborative protocol each vehicle stores the following information:

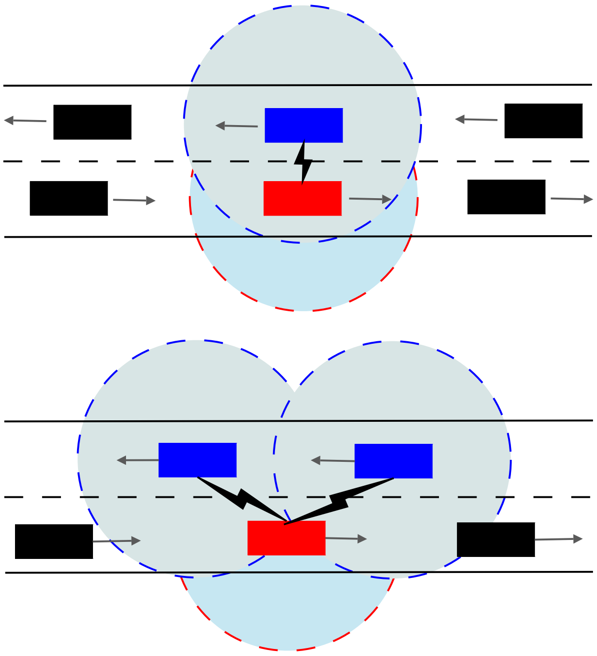

Figure 3.4: Single hop and Multi hop delivery in VANETs

1. a map of the area in which vehicle is moving (we used a map of Pisa in our experiments);
2. a table containing the most updated information about travelling times for every roadway in the map;
3. the information about travelling time of roadway in which vehicle moving on.

The table that stores travelling times is a collection of rows that have format shown in Table 3.2. The field `id` is an integer that identifies unequivocally a roadway inside the network. The field `travellingTime` contains the time needed to move from the start to the end of roadway identified by `id`. Finally, the field `timestamp` expresses *freshness* of the information about travelling time. In this case, a high value means a more recent information.

Given this network of nodes several different communication protocols are possible. At this stage we explore the *Simple Push Protocol*, that is described below.

id	travellingTime	timestamp
32 bits	32 bits	32 bits

Table 3.2: Scheme of information about travelling time.

3.4 The Simple Push Protocol

The key feature of the Simple Push Protocol (SPP) are summarize below:

1. the communication is single-hop, only two nodes within a distance R can communicate with each other. Thanks to the single hop communication no message routing is necessary;
2. the vehicles are not allowed to request information regarding a any travelling time;
3. based on some *transmission rule* each vehicle broadcasts to all the vehicles within a distance R the travelling times of the roads that are *relevant*.

Moreover, when the distance to the end of the roadway is smaller than a threshold value D, the vehicle recomputes the fastest path, if necessary. In this case the threshold value D must satisfy the following constraint:

$$vT_{\mathsf{dijkstra}} < D \tag{3.4}$$

where v is speed of vehicle and T_{dijkstra} is the run-time of Dijkstra's algorithm to find fastest path.

An interesting feature of this protocol is that the diffusion of the information is more efficient when the traffic is intense. This allows the vehicles to select faster and faster routes when it is more needed (i.e. in the condition of intense traffic). Conversely, when the traffic is low the protocol is inefficient because it relies in the capability of the vehicles to transport information. However in this case we don't expect that the fastest route differs significantly than the shortest route that is compute assumed that the roads along the path are empty.

In SPP, when the navigation system is started, it initializes the travelling times table with the estimated time $T_{\mathsf{estimated}}$, which is defined as the time required by a vehicle to move along the road in absence of traffic. Basically, it evaluates the motion under uniform acceleration using the classic linear equations [3]. Then the vehicle performs the Dijkstra's algorithm, using the travelling times as costs, to find the fastest path to the destination. During the trip the vehicle follows the rules of the protocol described previously. Specifically, when the vehicle has some *relevant* information it broadcasts them to the vehicles within a radius R. At the same time, it always listens for *relevant* information transmitted by other vehicles and it possibly stores the information in its own table.

Below we examine the rules that we adopt to determine when the travelling times of the roads are relevant or not.

3.4.1 Relevance of the information

The key aspect of the collaborative protocol is the *relevance* of the information, since it determines when some travelling times it transmitted or not. Clearly, all the vehicles could always transmit the complete table of all the traveling times. However, this behaviour leads to a waste in the communication channel. Instead, we choose to transmit only the information that can be relevant for the neighboring vehicles.

The relevance of the travelling time of one road id is represented by a number between 0 (irrelevant) and 1 (very relevant). The relevance is the product three factors:

1. $\alpha(\mathsf{id}) \in [0, 1]$, representing the *freshness*, of the road id;
2. $\beta(\mathsf{id}) \in [0, 1]$, representing the *geographical utility*;
3. $\gamma(\mathsf{id}) \in [0, 1]$, representing the *deviation* from the last value.

The travelling time of a the road id is broadcasted to the neighboring vehicles when

$$\alpha(\mathsf{id})\,\beta(\mathsf{id})\,\gamma(\mathsf{id}) \geq \lambda \tag{3.5}$$

where λ is a given relevance threshold. Notice that if we assume $\lambda = 0$ and $R = \infty$, then SPP becomes the same as the centralized protocol.

We know explain all the three factors that determine the relevance.

Freshness α The freshness $\alpha(\mathsf{id})$ of the travelling time of the road id is a number between 1 (new) and 0 (old). We model it by the following exponential function:

$$\alpha(\mathsf{id}) = \exp\left(-\frac{t - \mathsf{timestamp(id)}}{t_0}\right) \tag{3.6}$$

where t_0 is a constant that can be tuned to adjust the sensitivity to the freshness, t is the current time, and $\mathsf{timestamp(id)}$ is the absolute time when the travelling time of the road id was measured. The profile of the freshness is shown in Figure 3.5, assuming $t_0 = 500$.

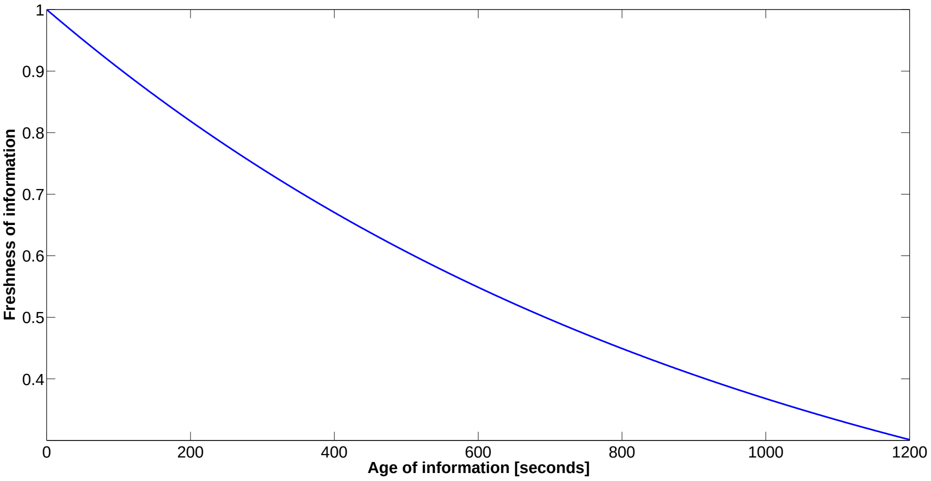

Figure 3.5: The freshness profile.

Geographical utility β The geographical utility $\beta(\mathsf{id})$ of the travelling time of the road id is modeled by a number between 1 (useful) and 0 (useless). It expresses *where* the travelling time of the road can help the other vehicles to find a fastest path.

Since it can be useless to broadcast the information about a road id that is too far from the sender, the geographical utility is a decreasing function of the distance between the position of the sender and the position of the road id. Hence we model it by the function

$$\beta(\mathsf{id}) = \exp\left(-\frac{\|x(\mathsf{sender}) - x(\mathsf{id})\|}{r}\right) \tag{3.7}$$

where $x(\mathsf{sender})$ is the current position of the vehicle, $x(\mathsf{id})$ is the position of the start of the road id, and r is a parameter that can be tuned to achieve the desired behaviour.

This concept is also represented in Figure 3.6.

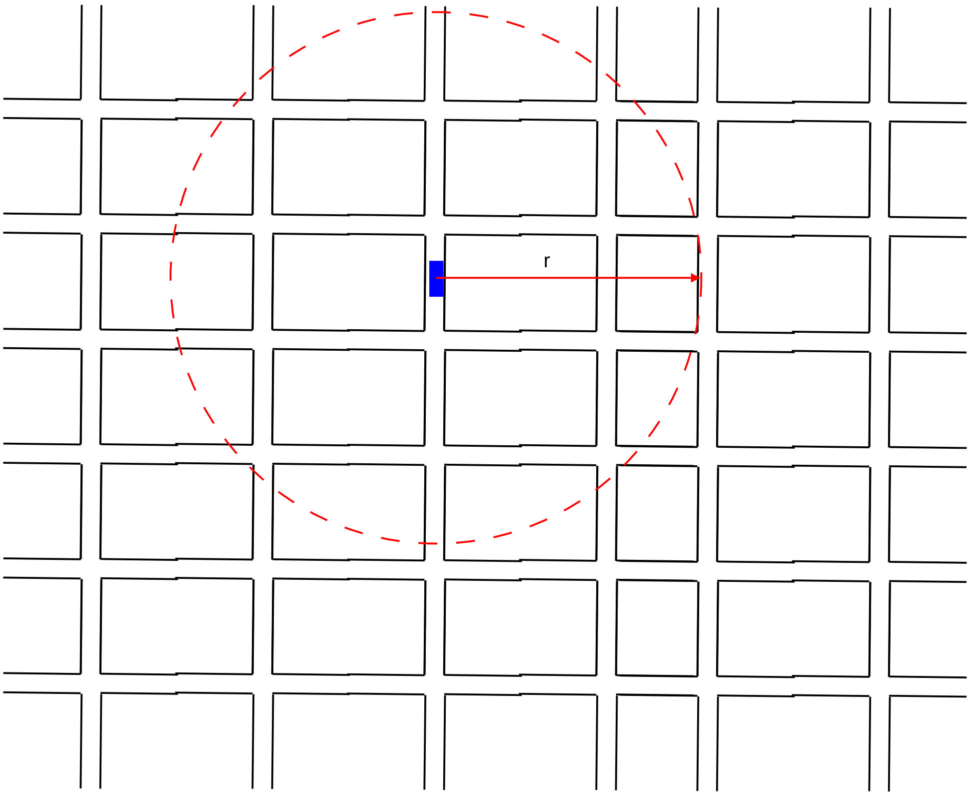

Figure 3.6: The geographical utility of information is function of radius r.

Deviation γ The deviation $\gamma(\mathsf{id})$ represents the difference between the travelling time measured by the vehicle and the travelling time estimated assuming standard traffic conditions. It assumes values between 1 (high difference) and 0 (no difference).

We introduce this weight because a travelling time is more important to transmit when the experienced value differs significantly than the foreseen value. On the other hand, if the travelling time does not differ than the estimated value there is no need to transmit it, and some medium bandwidth can be saved.

We choose to express it by:

$$\gamma(\mathsf{id}) = 1 - \exp\left(-\left(\frac{T_{\mathsf{measured}} - T_{\mathsf{estimated}}}{T_0}\right)^2\right) \tag{3.8}$$

where T_{measured} is the measured travelling time, $T_{\mathsf{estimated}}$ is the estimated travelling time, and T_0 can be tuned to adjust the sensitivity on the difference. The profile of γ is shown in Figure 3.7, in this case $T_0 = 10$.

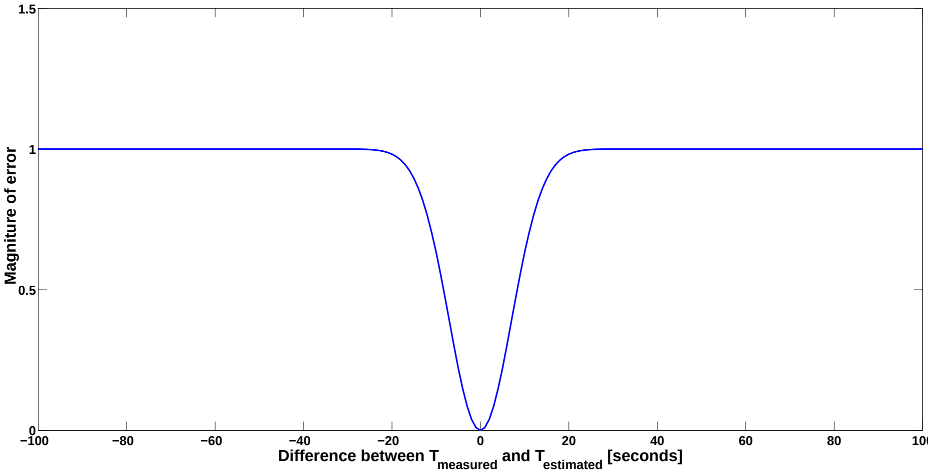

Figure 3.7: The magnitude of error profile

Chapter 4

Traffic Simulations

In this chapter we will show the results of the simulations. In the first section we will describe the scenario that we will use to analyze the traffic behaviour with the centralized and the collaborative protocols inside the simulator. In the following sections the results of the simulations are shown and they are analyzed in order to evaluate the performance of the protocols and the traffic behaviours.

4.1 Simulated scenario

All the simulations are performed using the same scenario, which is defined as follows:

1. a map of roadways system;
2. a group of vehicles that create traffic congestion;
3. a group of vehicles under analysis.

4.1.1 The map

We will use in all simulations a simplified map of Pisa in XML format. In creating this map we have adopted the following assumptions:

1. every road has one lane only;

2. all the lanes of every roads have the same width;
3. there are not traffic lights or traffic circles;
4. all the roads have the same speed limit.

Furthermore, we are not interested to simulate the full traffic behaviours, but we want to analyze the traffic behaviours on a particular path inside the map. We monitor the shortest path, which was found with Dijkstra's algorithm, from *S.S N. 224 di Marina di Pisa* to *SP2 Strada provinciale Calcesana, zona Ghezzano*. We choose these start and end points because they are far enough to allow a large number of alternative paths. Also we create traffic congestion on the shortest path because we want to show that the shortest path and the fastest path are not the same when there is traffic congestion on the roadways system.

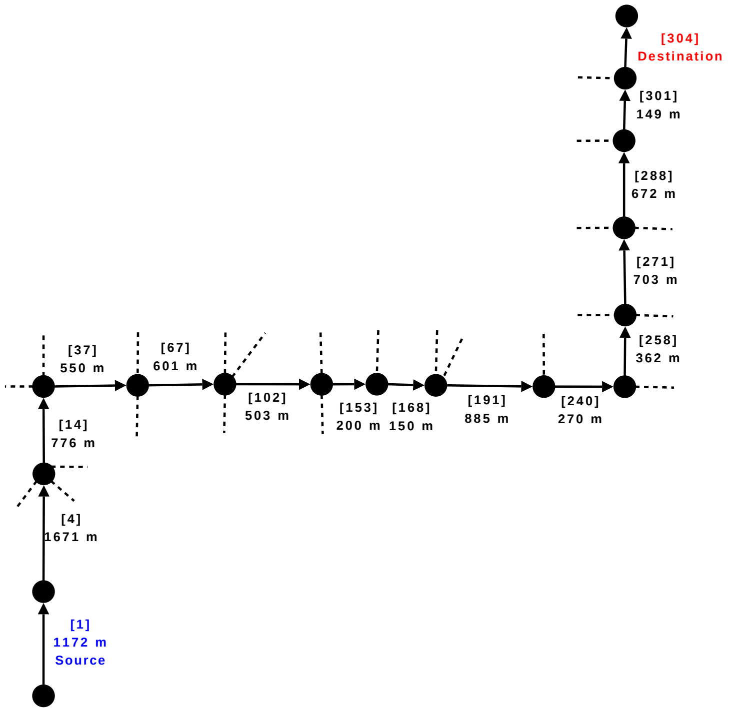

Figure 4.1: Graph which reproduce monitored path.

Figure 4.1.1 shows the path which we are using in the simulation. The dashed lines that exit from vertices of the graph, underline the presence of an alternative path to reach the destination.

4.1.2 The vehicles

In every simulation there are two groups of vehicles. The first, called G_{dummy}, is used to create traffic congestion on the roadways system: it is the same in all simulations and it is not under test. The second, called G_{test}, is the group of vehicles under monitoring: in every simulation we change the number of vehicles to analyze the situation under different traffic conditions.

We are interested in monitoring the average time required for a vehicle to get across its route:

$$T_{travel} = \Delta t = t_{end} - t_{start} \tag{4.1}$$

And we want to make T_{travel} as closest as possible to $\bar{T}$ which is the value of the average time when there is not traffic congestion.

Since we are interested in monitoring the average time, we make some simplifying assumptions, in both groups of vehicles, which ease the simulation, without altering the overall result.

Hence we assume that:

1. the vehicles have all the same length and the same width;
2. the vehicles can increase their speed up to the speed limit only;
3. the top speed is the same for all the vehicles in both groups;
4. the maximum acceleration and the maximum deceleration are the same for all the vehicles in the monitored group, but they are set to low values to create congestion in the unmonitored group;
5. a vehicles cannot overtake a slower one.

All the vehicles of the first group are equipped with a standard GPS navigator, so they follow the shortest path to reach their destination. The vehicles of the monitored group are equipped with a navigator which tries to make them reach their destinations through the fastest path, using centralized or collaborative protocol.

4.1.3 Traffic congestion on monitored path

Before simulating the behavior of the communication protocols, we try to analyze how the traffic congestion changes on the monitored paths. Table 4.1 shows the time required for a vehicle to get across the monitored path when there are not vehicles on it.

L_{path}	T_{travel}	v_{avg}
8664 m	657 s	13.19 m/s

Table 4.1: Path length L_{path}, travelling time T_{travel} and average speed v_{avg} of the path without traffic congestion.

When we increase the number of vehicles in the roadways system also the traffic congestion and travelling time increase as expected. Variation of average time with traffic congestion is listed in Table 4.2: We set the number of vehicles to 1300 in the unmonitored group, for all the simulations of the communication protocols. This is a good trade off between the level of traffic congestion and the speed of the simulations.

Vehicles on roads	T_{travel}	v_{avg}
1	657 s	13.19 m/s
150	1725 s	5.02 m/s
1300	2068 s	4.19 m/s
2600	2160 s	4.01 m/s

Table 4.2: Statistics of the path with traffic congestion.

4.2 Simulations of centralized protocol

In this section we proceed in analyzing the results of the simulations, that we made on the centralized protocol. The vehicles of the monitored group are equipped with the same communication system which implements centralized protocol and they have the same characteristics. In each simulation we change the number of vehicles and their positions in the roadways system to investigate how the performance of the protocol changes.

4.2.1 Test 1

Through this test we want to show that when there is no information about traffic congestion, the algorithm does not produce the fastest path, which has the minimum travelling time, but the shortest path, which has the minimum distance. This happens because there are not enough vehicles with this type of navigation system, therefore it is impossible to find the fastest path.

The scenario In this test there are two groups of vehicles, as shown in Table 4.3.

	Vehicles	Max speed
G_{dummy}	1300	5 m/s
G_{test}	1	14 m/s

Table 4.3: The groups of vehicles in test 1.

The G_{dummy} group of vehicles creates traffic congestion on the monitored path, as previously explained.

The G_{test} group of vehicles uses the centralized protocol version of the communication system and it is the group of vehicles under test. In this case only one vehicle belongs to G_{test} because we want to analyze the behaviour of the communication system when there is not information about the traffic congestion.

The evolution The vehicles in G_{dummy} are loaded at value 0 s of the simulation time and they are moved along the path to create the traffic congestion.

The vehicle in G_{test} is loaded at value 405 s of the simulation time. It cannot find the fastest path, since there are not information about the traffic congestion. Therefore it uses the shortest path to reach the destination instead of the fastest one. The vehicle ends its trip at value 2473 s of the simulation time.

The speed profile of the vehicle is shown in Figure 4.2.1. As expected, the vehicle moves to the congested path, it reaches the traffic queue at value 415 s of the simulation time and then it moves to the destination at speed of 5 m/s approximately.

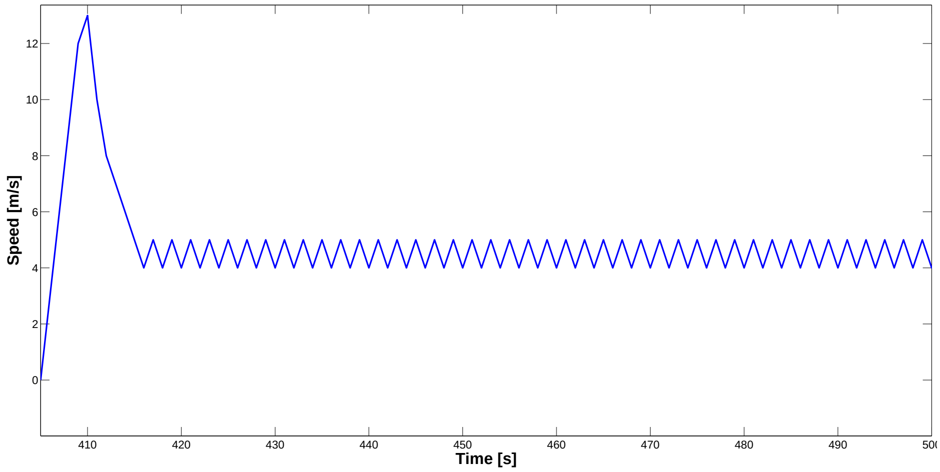

Figure 4.2: The speed profile of the vehicle in G_{test}.

The results As expected the route of the vehicle in G_{test} is equal to the route of the vehicles in G_{dummy}. This happens because the vehicle in G_{test} has no information about travelling times of the roads in the map, so the Dijkstra's algorithm uses road lengths as inputs instead of their corresponding travelling times. The results of the simulation are shown in Table 4.4.

L_{path}	T_{travel}	v_{avg}	v_{max}
8664 m	2068 s	4 m/s	13 m/s

Table 4.4: The results of test 1.

4.2.2 Test 2

In this test we want to investigate how many vehicles must move along the monitored path to detect the traffic congestion. Therefore we increase the number of vehicles in the monitored group until this detection happens: this is true whenever at least one vehicle changes its path during the simulation.

The scenario In this test the groups of vehicles are two as in the previous test (see Table 4.5), but the number of vehicles in G_{test} is not fixed.

	Vehicles	Max speed
G_{dummy}	1300	$5\ m/s$
G_{test}	$\{2,N\}$	$14\ m/s$

Table 4.5: The groups of vehicles in test 2.

The group of vehicles G_{dummy} is used to create traffic congestion.

The group of vehicles G_{test} uses the centralized protocol version of the communication system and it is the group of vehicles under test. In this case the number of vehicles in G_{test} is not fixed but it is increased after each simulation until the traffic congestion is detected.

The evolution The vehicles in G_{dummy} are loaded at value $0\ s$ of the simulation time and they are moved along the path to create the traffic congestion.

Vehicle	t_{start}	t_{end}	T_{travel}
1	$405\ s$	$2473\ s$	$2068\ s$
57	$741\ s$	$2053\ s$	$1312\ s$

Table 4.6: Start and end times of the first vehicle and the last one in G_{test}.

The first vehicle in G_{test} is loaded at value $405\ s$ of the simulation time and a new vehicle is loaded every 6 seconds. The traffic congestion is not detected as long as the number of vehicles that belong to G_{test} is lower than 57, so the vehicles from 1 to 56 use the shortest path instead of the fastest one and they move to the traffic queue, increasing the traffic congestion.

When the number of the vehicles reaches 57, the vehicles from 1 to 56 have the same behaviour of the previous simulation, but the last vehicle does not move to the traffic jam. It uses information sent by other vehicles using the centralized protocol and it finds the fastest path. Therefore the vehicle

57 is loaded at 741 *s* of the simulation time but it reaches the destination before vehicle 1, as shown in Table 4.6.

This is the behaviour of all vehicles after the 57^{th} if we increase the number of vehicles in G_{test} up to 91. The next test will investigate what happens to the traffic behaviour when the number of vehicles in G_{test} is 92.

The results In the simulated scenario the centralized protocol needs at least 741 *s* to detect the traffic congestion. The vehicles that are loaded into the roadways system during this interval cannot use any information about the traffic status, therefore they move along the congested path to reach the destination and they allow gathering information about the status of the traffic congestion. When this information is gathered the next vehicles can exploit it to reach the destination using the fastest path. In Table 4.7 the information about two different paths that the vehicles use to reach the destination is shown, while Figure 4.2.2 shows how the travelling times vary when the number of vehicles in G_{test} increases from 1 up to 91.

Vehicles	L_{path}	T_{travel}	v_{avg}	v_{max}
{1,56}	8664 m	2070 s	4.19 m/s	12.98 m/s
{57,91}	8961 m	1312 s	6.82 m/s	14 m/s

Table 4.7: The paths used by the vehicles in the G_{test}.

4.2.3 Test 3

Through this test we want to show that even young vehicles can help long lived ones. This happens when the young vehicles use the information provided by the protocol to find an alternative path: it is possible that the information about the traffic congestion is not complete, so the algorithm which uses this information is able to find an alternative path with some roads in common with the congested path. Therefore the vehicles that use this alternative path overtake the ones that are slowed down by the traffic congestion. When the young vehicles move along the common roads they collect information about traffic congestion of these roads and they help the other vehicles to find an alternative path.

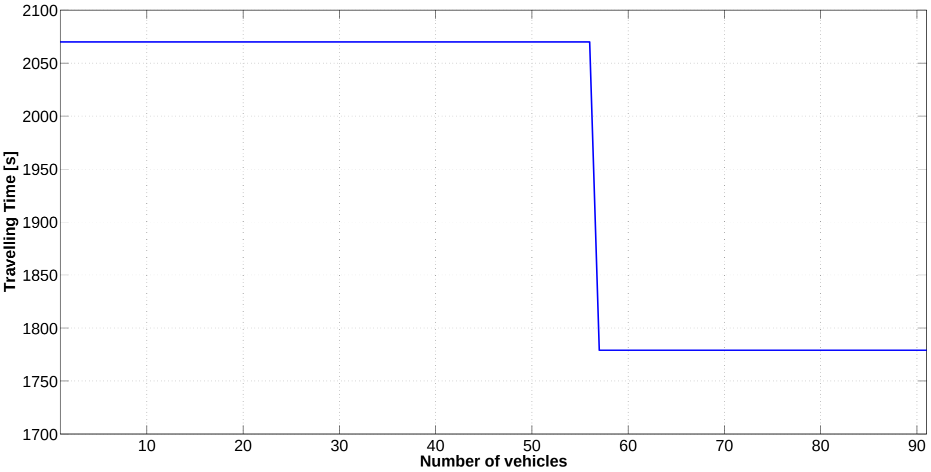

Figure 4.3: How the travelling time changes when the number of vehicles in G_{test} increases.

The scenario In this test the groups of vehicles are two as shown in Table 4.8.

	Vehicles	Max speed
G_{dummy}	1300	5 m/s
G_{test}	92	14 m/s

Table 4.8: The groups of vehicles in test 3.

The group of vehicles G_{dummy} is the same of the previous test and it is used to creates traffic congestion.

The group of vehicles G_{test} has the centralized protocol version of the communication system and it is the group of vehicles under test. In this case the number of vehicles in G_{test} is fixed and it is set to 92. This particular value is discovered empirically during the simulation phase.

The evolution The vehicles in G_{dummy} are loaded at value 0 s of the simulation time and they are moved along the path to create the traffic congestion.

The first vehicle in G_{test} is loaded at value 405 s of the simulation time and a new vehicle is loaded every 6 seconds. Like the previous test, the first

57 vehicles do not recognize the traffic congestion then they try to reach the destination using the shortest path. The other vehicles use the information about the traffic status to find the fastest path. In this particular case the last vehicle, which is labelled as 92, finds a new path using the information provided by the vehicles from 57 to 91. Therefore it moves faster than the other vehicles through this new path and it overtakes all the vehicles.

This new path is different from the monitored path used by the vehicles from 1 to 56, only for few roads, therefore the vehicle number 92 moves along the new roads and it overtakes the other vehicles, but soon after it moves again along the roads that belong to the monitored path so it helps the others to change their routes to avoid the congested path.

The results The experiments on the traffic behaviour described in the previous paragraph show that there are significant differences in the simulation results, as shown in Table 4.9.

The vehicles use many alternative paths, in this case they are 27. Therefore they move along a large number of roads and they use the roadways system more efficiently.

An important effect of this more efficient use of the roadways system is a significant decrease in the T_{travel} of all vehicles. In this case the average T_{travel} is 1482.60 s while in the previous test it was about 1780 s.

4.2.4 Test 4

In this test we want to investigate the behaviour of the vehicles that use the centralized protocol when the information about the traffic congestion is available and up-to-date.

The scenario There are three groups of vehicles instead of two, as shown in Table 4.10.

The group of vehicles G_{dummy} creates traffic congestion on the monitored path, as it does in the previous tests.

The group of vehicles G_{mule} uses the centralized protocol version of the

Path	Vehicles	L_{path}	T_{travel}	v_{avg}
p_1	9	8961.40 m	1556.78 s	5.76 m/s
p_2	7	9093.00 m	1588.71 s	5.72 m/s
p_3	7	9167.10 m	1972.29 s	4.65 m/s
p_4	6	9126.20 m	1525.83 s	5.98 m/s
p_5	2	9427.60 m	1441.00 s	6.54 m/s
p_6	1	9633.40 m	1925.00 s	5.00 m/s
p_7	2	9559.30 m	1543.00 s	6.20 m/s
p_8	4	9592.50 m	1401.50 s	6.84 m/s
p_9	8	9952.40 m	1337.13 s	7.44 m/s
p_{10}	1	9978.20 m	1348.00 s	7.40 m/s
p_{11}	4	10110.0 m	1384.75 s	7.30 m/s
p_{12}	3	10746.0 m	1741.33 s	6.17 m/s
p_{13}	2	10720.0 m	1769.00 s	6.06 m/s
p_{14}	2	9826.40 m	1403.00 s	7.00 m/s
p_{15}	1	9734.10 m	1554.00 s	6.26 m/s
p_{16}	5	10442.0 m	1445.20 s	7.23 m/s
p_{17}	3	10102.0 m	1452.00 s	6.96 m/s
p_{18}	1	10117.0 m	1251.00 s	8.09 m/s
p_{19}	2	9540.10 m	1181.00 s	8.08 m/s
p_{20}	2	9565.90 m	1196.00 s	8.00 m/s
p_{21}	4	9439.90 m	1229.00 s	7.68 m/s
p_{22}	3	9889.00 m	1590.67 s	6.22 m/s
p_{23}	3	9863.20 m	1604.00 s	6.15 m/s
p_{24}	2	9414.10 m	1235.00 s	7.62 m/s
p_{25}	4	9704.90 m	1155.00 s	8.40 m/s
p_{26}	3	8469.90 m	1155.00 s	7.33 m/s
p_{27}	1	8781.30 m	1458.00 s	6.02 m/s

Table 4.9: The paths used by the vehicles in the G_{test}.

	Vehicles	Max speed
G_{dummy}	1300	5 m/s
G_{mule}	10	14 m/s
G_{test}	1	14 m/s

Table 4.10: The groups of vehicles in Test 4.

communication system and it simulates the behaviours of the local transport, for example the buses or taxis. The vehicles that belong to G_{mule} move along the path under test at regular intervals and they collect information about the travelling times of the roads that belong to the path.

The group of vehicles G_{test}, which has the same communication system of the previous one, is the group of vehicles under test. In this case only one vehicle belongs to G_{test} because we are interested in monitoring only one path. It begins its trip when the information about travelling times of the roads inside the path are already collected by the previous group, so the Dijkstra's algorithm uses this information to find the fastest path.

The evolution The vehicles in G_{dummy} are loaded at value 0 s of the simulation time and they are moved along the path to create the traffic congestion.

The vehicles in G_{mule} are loaded at value 9 s of the simulation time and they start to move along the monitored path to collect the traffic information.

The vehicle in G_{test} is loaded at value 413 s of the simulation time. At this time the vehicles in G_{mule} moved along the monitored path about 3 times and all information about the traffic congestion is available to the vehicle in G_{test}, therefore its navigation system can use it to find the fastest path instead of the shortest one.

The results As expected the vehicle in G_{test} uses the information collected by the vehicles in G_{mule} to minimize the travelling time. The results of the simulation are shown in Table 4.11.

L_{path}	T_{travel}	v_{avg}
9024 m	1038 s	8.69 m/s

Table 4.11: The results of test 4.

We can see that the availability and the freshness of the information allow a 50% reduction of T_{travel} with respect to the worst case, which happens when the vehicle uses the shortest path.

4.2.5 Conclusions

The previously explained tests are just a small part of all the experiments that we have conducted during the simulation phase. We have performed many tests to investigate the behaviour of the centralized protocol in different scenarios. We can summarize the results as follows:

1. the time which is needed by the protocol to detect the traffic congestion is based on the time which is needed by the first vehicle to move along the congested path, therefore the congestion on short paths is detected before the congestion on long paths;

2. in a few cases the protocol is able to detect the traffic congestion on one or more roads even if only one vehicle moves along that/those roads, but the traffic congestion can be better avoided if a large number of vehicles move along the same roads. This number depends on the characteristic of the monitored path;

3. in a scenario in which there is traffic congestion on the path from A to B, the lower bound of T_{travel} is not the travelling time of that path when there is no congestion, whereas it is the minimum travelling time of the alternative paths.

4.3 Simulations of collaborative protocol

In this section we analyze the results of the simulations on the collaborative protocol version of the communication system. As the previous simulations, the vehicles of the monitored group are equipped with the same communication system which implements the collaborative protocol. In this case there is a new simulation variable: the communication radius R, which is set to 100 m and it remains the same in all simulations. Whenever it is possible the performances of this protocol will be compared to the performances of the centralized protocol version.

4.3.1 Test 1

In this test we use the same approach which is used in the test of the centralized protocol: a group of vehicles produce the traffic congestion on the monitored path, while further vehicles are loaded at regular intervals to produce the flow of the vehicles under test.

We want to show that:

1. it is possible to create an inter-vehicle network from the first vehicle of the flow to the last one, if the distance between the vehicles inside the flow of the vehicles is not greater than the communication radius R;
2. the collaborative protocol can use this network to exchange information about the traffic congestion in order to compute the fastest path.

Furthermore, we are interested in analyzing the time this protocol takes to detect the traffic congestion.

The scenario There are two groups of vehicles, as shown in Table 4.12.

	Vehicles	Max speed
G_{dummy}	1300	5 m/s
G_{test}	{1, N}	14 m/s

Table 4.12: The groups of vehicles in test 1.

The group of vehicles G_{dummy} creates traffic congestion on the monitored path, as the previous tests.

The group G_{test} consists of vehicles using the collaborative protocol version of the communication system. This is the group of vehicles under test and we increase the number of units that belong to it in every simulation.

The first vehicle in G_{test} is loaded at value 405 s of the simulation time and the other vehicles are loaded every 6 s. We can see that the distance between two vehicles is about 50 m, which is lower than the communication radius R. This fact makes possible to establish a network connection between them, therefore a multihop wireless network links the first vehicle of G_{test} and the last vehicle loaded into the monitored path. as shown in Figure 4.3.1.

The evolution The vehicles in G_{dummy} are loaded at value 0 s of the simulation time and they are moved along the path to create the traffic congestion.

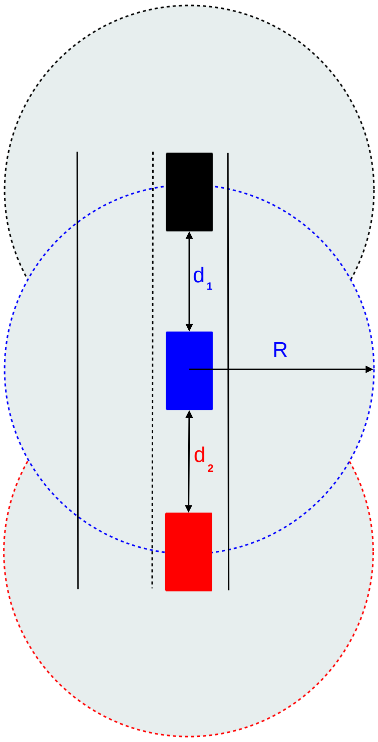

Figure 4.4: The inter-vehicle network can be established if the distance d_i is lower than communication radius R.

The vehicles that are loaded later are connected to the inter-vehicle network and they receive the information about the traffic congestion. This information is sent by the vehicles at the head of the network and it propagates through it.

The first vehicle which uses information about the traffic congestion to find the fastest path is the vehicle number 75. It is loaded at 849 s of the simulation time and it reaches the destination at 2090 s of the simulation time, with the T_{travel} equal to 1241 s. The first vehicle reaches the destination at 2473 s of the simulation time, with the T_{travel} equal to 2068 s.

The results The results of the simulations are shown in Table 4.13.

Vehicles	L_{path}	T_{travel}	v_{avg}	v_{max}
$\{1, 74\}$	$8664\ m$	$2070\ s$	$4.19\ m/s$	$12.98\ m/s$
$\{75, 100\}$	$8961\ m$	$1250\ s$	$7.17\ m/s$	$14\ m/s$

Table 4.13: The results of the test 1.

Comparing the results of the simulations on the collaborative protocol with the ones on the centralized version previously described, we can see that:

1. the collaborative protocol is slower than the centralized protocol to detect the traffic congestion, because the information must be propagated through the network to reach the other vehicles;

2. the collaborative protocol has less efficient use of the roadways system, because the data dissemination produces a lower number of alternative paths than the centralized protocol.

4.3.2 Test 2

In this test we want to investigate the behaviour of the collaborative protocol when:

1. any inter-vehicle network cannot be established inside the same flow of vehicles because the distance between two of them is greater than the communication radius R;

2. the previous flow meets another one, with the same characteristics, which moves in the opposite direction.

The scenario There are three groups of vehicles, as shown in Table 4.14. A group of vehicles produce the traffic congestion on the monitored path, while the others create two flows of vehicles provided with the communication system, that move in opposite directions:

1. the first flow is comprised of the vehicles under test, that move from the source position to the destination. In this case the distance between

these vehicles is greater than the communication radius R therefore a multihop network among the vehicles cannot be established;

2. the second flow contains the vehicles that are used to propagate data. They move along the monitored path but in the opposite direction with respect to the previous flow.

	Vehicles	Max speed
G_{dummy}	1300	$5\ m/s$
G_{mule}	1	$v_{max}\ m/s$
G_{test}	2	$14\ m/s$

Table 4.14: The groups of vehicles in test 2.

The group of vehicles G_{dummy} creates traffic congestion on the monitored path, as in the previous tests.

The group G_{mule} consists of the vehicles using the collaborative protocol version of the communication system. This group of vehicles is not under test and it is used to create inter-vehicle networks together with the vehicles of the other group. We use only one vehicle for this group and we will change its maximum speed v_{max} to investigate how the behaviour of the protocol changes accordingly.

The group G_{test} consists of the vehicles that use the collaborative protocol version of the communication system. This is the group of vehicles under test and no inter-vehicle network can be established because the distance between two of them is greater than the communication radius R. We use two vehicles, the first is the vehicle which moves along the monitored path and it collects the information about the traffic congestion. The second is the vehicle which establishes a connection with the vehicle in G_{mule} to get the information about the traffic congestion and then it computes the fastest path.

The evolution The vehicles in G_{dummy} are loaded at value $0\ s$ of the simulation time and they are moved along the path to create the traffic congestion.

The first vehicle of G_{test} is loaded at value 125 s of the simulation time. It moves along the monitored path because no information about the traffic congestion is available. The second vehicle of G_{test} is loaded at value 636 s of the simulation time. It cannot establish any connection with the previous vehicle, therefore it cannot receive the information about the traffic congestion and then it moves along the monitored path as the previous vehicle.

The vehicle in G_{mule} is loaded at value 0 s of the simulation time and it starts to move in the same direction as the first vehicle of G_{test}. The value of v_{max} is set to 5 m/s.

Figure 4.3.2 shows what happens when the vehicle of G_{mule} meets the other vehicle: (a) when the vehicle in G_{mule} meets the first vehicle of G_{test} an inter-vehicle network is established between them and they exchange the information about the traffic congestion. (b) The vehicle in G_{mule} moves along its path. (c) When the vehicle in G_{mule} meets the other vehicle of G_{test} it establishes an inter-vehicle network and it sends the information about the traffic congestion, previously collected.

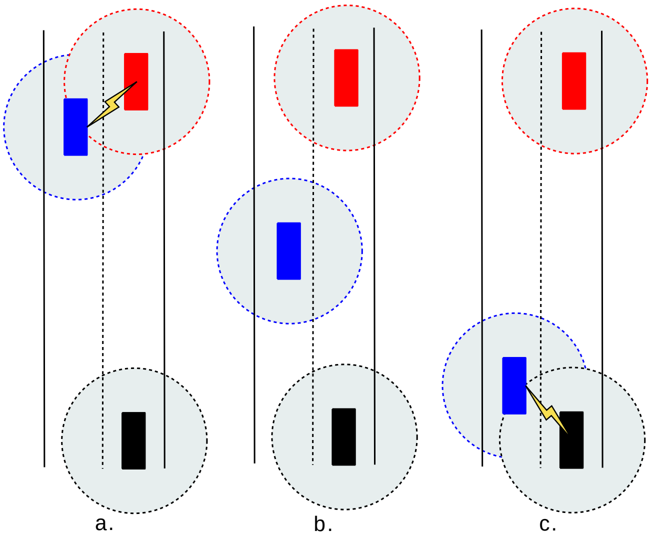

Figure 4.5: The inter-vehicle networks that are established between vehicles.

This behaviour allows the second vehicle in G_{test} to change its route.

The results The results of the simulations are shown in Table 4.15.

Vehicle ID	L_{path}	T_{travel}	v_{avg}	v_{max}
1	$8664\ m$	$2104\ s$	$4.2\ m/s$	$12.98\ m/s$
2	$8781\ m$	$1496\ s$	$5.9\ m/s$	$14.00\ m/s$

Table 4.15: The results of the test 2.

As expected, the vehicle in G_{mule} makes the communication between vehicles in G_{test} possible, although not through a direct link. By doing so, the first vehicle in G_{test} shares information about traffic congestion with the second one. Therefore the second vehicle is able to find an alternative path and it reaches the destination before the first vehicle. We notice that T_{travel} is function of time to propagate the information, which depends on the speed (v) of the vehicle in G_{mule}, as follows:

1. if we decrease v then T_{travel} increase.
2. if we increase v then T_{travel} decrease.

In this scenario the protocol does not work properly if the value of v:

1. is grater than threshold v_{max}. In this case the protocol cannot work because the high speed of the vehicle does not allow transmitting of the information. v_{max} depends on the time which is needed to complete transmission of the information.
2. is smaller than threshold v_{min}. In this case the communication happens, but the information is useless because it is too old. v_{min} depends on the distance between two vehicles and on the time which information needs to become old (*freshness of the information*).

4.3.3 Conclusions

We have performed many tests to investigate the behaviour of the collaborative protocol in different scenarios. We can summarize the results as follow:

1. the collaborative protocol takes advantage of the networks getting established among the vehicles that move in the same direction as well as among the vehicles that move in the opposite direction;
2. there is a gap between the time when the information is produced and the time when the information is used. This gap affects the performances of the collaborative protocol;
3. on average the collaborative protocol needs much more vehicles than the centralized protocol to work properly;
4. the centralized protocol has bigger decrease of T_{travel} than the collaborative protocols on average.

Bibliography

[1] Bonnmotion, a mobility scenario generation and analysis tool.

[2] Canu project home page.

[3] Equations of motion. http://en.wikipedia.org/wiki/Equations_of_Motion.

[4] Jist/swans. http://jist.ece.cornell.edu/index.html.

[5] The network simulator ns-2. http://www.isi.edu/nsnam/ns.

[6] Vissim. http: //www.english.ptv.de/cgi-bin/traffic/traf_vissim.pl.

[7] Saha A. and Johnson D. Modelling mobility for vehicular ad-hoc networks. In *ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, 2004.

[8] D. B. Johnson A. K. Saha. Modeling mobility for vehicular ad hoc networks. In *1st ACM Workshop on Vehicular Ad Hoc Networks (VANET)*, 2004.

[9] K. Gopalan A. Wang A. Mahajan, N. Potnis. Evaluation of mobility models for vehicular ad-hoc network simulations. In *International Workshop on Next Generation Wireless Networks (WoNGeN)*, 2006.

[10] S. Das M. Gerla L. Kleinrock A. Nandan, S. Tewari. Adtorrent: Delivering location cognizant advertisements to car networks. In *IEEE/IFIP Annual Conference on Wireless Ondemand Network Systems and Services (WONS)*, 2006.

[11] K. I. Ahmed. Modeling drivers acceleration and lane changing behaviors. 1999.

[12] M. Gerla B. Zhou, K. Xu. Group and swarm mobility models for ad hoc network scenarios using virtual tracks. In *IEEE Military Communications Conference (MILCOM)*, 2004.

[13] Fan Bai and Ahmed Helmy. A survey of mobility models in wireless ad-hoc networks. University of Southern California, USA, 2004.

[14] R. Baumann. Vehicular ad hoc networks (vanet). 2004.

[15] C. Bettstetter and C. Wagner. The spatial node distribution of the random waypoint mobility model. In *in Proc. German Workshop on Mobile Ad-Hoc Networks (WMAN), Ulm,Germany, GI Lecture Notes in Informatics, no. P-11*, pages 41–58, 2002.

[16] J. Blum, A. Eskandarian, and L.J. Hoffman. Challenges of intervehicle adhoc networks. *IEEE Transaction on Intelligent Transportation Systems*, 5(4):347–351, 2004.

[17] L. Breisemeister. Group membership and communication in highly mobile ad-hoc networks. School of Electrical and Computing Science, Tecnhical University of Berlin, 2001.

[18] U.S. Census Bureau. Topologically integrated geographic encoding and referencing (tiger) system.

[19] R. Diaconescu V. Cristea L. Ifode C. Gorgorin, V. Gradinescu. An integrated vehicular and network simulator for vehicular ad-hoc networks. In *European Simulation and Modelling Conference (ESM)*, 2006.

[20] Turner-Fairbank Highway Research Center. Revised monograph on traffic flow theory. 2001.

[21] J. Broch D. B. Johnson, D. A. Maltz. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *Ad Hoc Networking*, pages 139–172, 2001.

[22] R. W. Rothery D. C. Gazis, R. Herman. Nonlinear follow-the-leader models of traffic flow. pages 545–567, 1961.

[23] C. Rossel P. Wagner D. Krajzewicz, G. Hertkorn. Sumo (simulation of urban mobility): An open-source traffic simulation. In *Middle East Symposium on Simulation and Modelling (MESM)*, 2002.

[24] F. E. Bustamante D. R. Choffnes. An integrated mobility and traffic model for vehicular wireless networks. In *2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, 2005.

[25] V. Davies. Evaluating mobility models within an ad hoc network. 2000.

[26] E. W. Dijkstra. A note on two problems in connexion with graphs. In *Numerische Mathematik, 1*, page 269271, 1959.

[27] A. Einstein. Investigations on the theory of the brownian motion. In *Einstein, Collected Papers, vol. 2*, pages 170–82, 206–22, 1926.

[28] Ralph Johnson John M. Vlissides Erich Gamma, Richard Helm. In *Design Patterns: Elements of Reusable Object-Oriented Software*, 1995.

[29] R. Mangharam et al. Groovesim: a topography-accurate simulator for geographic routing in vehicular networks. In *2nd ACM Workshop on Vehicular Ad Hoc Networks*, 2005.

[30] Universal Mobile Telecommunication System (UMTS) ETSI. Selection procedures for the choice of radio transmission technologies of the umts. In *UMTS 30.03 Version 3.2.0*, 1998-2004.

[31] A. Helmy F. Bai, N. Sadagopan. The important framework for analyzing the impact of mobility on performance of routing protocols for adhoc networks. In *EAdHoc Networks Journal - Elsevier Science, Vol. 1, Issue 4*, pages 383–403, 2003.

[32] K.-C. Lan F. Karnadi, Z. Mo. Rapid generation of realistic mobility models for vanet. In *11th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2005.

[33] M. Dias de Amorim S. Fdida F. Legendre, V. Borrel. Modeling mobility with behavioral rules: the case of incident and emergency situations. In *Asian Internet Engineering Conference (AINTEC)*, 2006.

[34] M. Dias de Amorim S. Fdida F. Legendre, V. Borrel. Reconsidering microscopic mobility modeling for self-organizing networks. In *IEEE Network Magazine*, 2006.

[35] C. Gawron. Simulation-based traffic assignment. 1998.

[36] P. Hidas. A car-following model for urban traffic simulation. In *Traffic Engineering and Control, vol. 39 no 5*, pages 300–305, 1998.

[37] D. B. Johnson Y.-C. Hu J. Broch, D. A. Maltz and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *in Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking(Mobicom98)*, 1998.

[38] F. Filali C. Bonnet J. Haerri, M. Fiore. Vanetmobisim: generating realistic mobility patterns for vanets. In *ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, 2006.

[39] S. Bohacek J. Kim, V. Sridhara. Realistic simulation of urban mesh networks - part i: Urban mobility. In *Technical Report, Univeristy of Delaware*, 2006.

[40] B. Srinivasan J. P. Singh, N. Bambos and D. Clawin. Wireless lan performance under varied stress conditions in vehicular traffic scenarios. In *In IEEE VTC 2002 Fall, volume 2, pages 743747*, 2002.

[41] Fethi Filali Jerome Harri and Christian Bonnet. Mobility models for vehicular ad hoc networks: A survey and taxonomy. Institut Eurecom Department of Mobile Communications, France, 2006.

[42] M. Schreckenberg K. Nagel. A cellular automaton model for freeway traffic. In *Journal de Physique I, vol. 1992, no. 2*, pages 2221–2229, 1992.

[43] P. Wagner P. Simon K. Nagel, D. E. Wolf. Two-lane traffic rules for cellular automata: A systematic approach. 1997.

[44] S. Krauß. Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics. 1998.

[45] M. Bertini E. Croci L. Bononi, M. Di Felice. Parallel and distributed simulation of wireless vehicular ad hoc networks. In *ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2006.

[46] H. Rohling M. Lott L. Wischhof, A. Ebner and R. Halfmann. Sotis a self-organizing traffic information system. In *IEEE Semiannual Vehicular Technology Conference (VTC)*, 2003.

[47] W. Leutzbach. Introduction to the theory of traffic flow. 1988.

[48] Fellendorf M. and Vortisch P. Validation of the microscopic traffic flow model vissim in different real-world situations. 2000.

[49] R. Benekohal M. Aycin. Comparison of car-following models for simulation. In *Transportation Research Record No. 1678, TRB, National Research Council*, pages 116–127, 1999.

[50] M. Lott M. Rudack, M. Meincke. On the dynamics of ad hoc networks for inter-vehicle communications (ivc). In *International Conference on Wireless Networks (ICWN)*, 2002.

[51] D. Helbing M. Treiber. Realistische mikrosimulation von strassenverkehr mit einem einfachen modell. In *Symposium Simulationstechnik (ASIM)*, 2002.

[52] D. Helbing M. Trieber, A. Hennecke. Congested traffic states in empirical observations and microscopic simulations. In *Phys. Rev. E 62, Issue 2*, 2000.

[53] L. B. Michael and M. Nakagawa. Non-platoon inter-vehicle communication using multiple hops. In *IEICE Trans. Commun, E82-B(10)*, 1999.

[54] S. Ray M.J. Feeley, N.C. Hutchinson. Realistic mobility for mobile ad hoc network simulation. In *Lecture Notes in Computer Science (LNCS), Vol. 3158*, pages 324–329, 2004.

[55] G.B. Whitham M.J. Lighthill. On kinematic waves ii: A theory of traffic flow on long crowded roads. pages 317–345, 1955.

[56] K. Nagel N. Cetin, A. Burri. A large-scale multi-agent traffic microsimulation based on queue model. In *Swiss Transport Research Conference (STRC)*, 2003.

[57] L. Wolf S. Jaap, M. Bechler. Evaluation of routing protocols for vehicular ad hoc networks in city traffic scenarios. In *5th International Conference on Intelligent Transportation Systems Telecommunications (ITST)*, 2005.

[58] C. Gawron S. Krauß, P. Wagner. Metastable states in a microscopic model of traffic flow. In *Physical Review E, vol. 55, no. 304*, pages 55–97, 1997.

[59] H. Dia S. Panwai. Comparative evaluation of microscopic car-following behavior. In *IEEE Transactions on Intelligent Transportation Systems, vol. 6, Issue 3*, page 314325, 2005.

[60] T. Fliess T. Schulze. Urban traffic simulation with psycho-physical vehicle-following models. In *Winter Simulation Conference (WSC)*, 1997.

[61] M. Takai, J. Martin, and R. Bagrodia. Effects of wireless physical layer modeling in mobile ad-hoc networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, 2001.

[62] Jeff Boleng Tracy Camp and Vanessa Davies. A survey of mobility models for ad hoc network research. In *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5*, pages 483–502, 2002.

[63] A. Uchiyama. Mobile ad-hoc network simulator based on realistic behaviour model. In *Demo session in MobiHoc2005*, 2005.

[64] T. Gross V. Naumov, R. Baumann. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2006.

[65] R. Wiedemann. Modelling of rti-elements on multi-lane roads. In *Advanced Telematics in Road Transport, Comission of the European Community, DG XIII*, 1991.

[66] G. Pei C.-C. Chiang X. Hong, M. Gerla. A group mobility model for ad hoc wireless networks. In *ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile System (MSWiM)*, 1999.

[67] Nitin H. Vaidya Xue Yang. A vehicle-to-vehicle communication protocol for cooperative collision warning. 2004.

[68] Y. Zhang. Scalability of car-following and lane-changing models in microscopic traffic simulation systems. Louisiana State University, 2002.

[69] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 125–138, Boston, MA, USA, 2004.