 DOI 10.1515/pik-2014-0005 — PIK 2014; 37(2): 81–92

Alexander von Bodisco* and Thomas Oberwallner

Different Strokes for Different Folks: An Evaluation of Road Networks and their Impact on VANET Mobility

Abstract: The majority of communication protocols for Vehicular Ad-hoc Networks (VANETs) have only been verified via simulations which used synthetic mobility models such as random waypoint or random direction. It is clear that such models do not reflect the specific nature of car movement. Specialized mobility models which try to model all characteristics of realistic car movement, e.g. different sizes and car types, driver behaviour, driving lanes, acceleration and traffic lights, provide higher accuracy but are usually too detailed to be practicable for network simulations.

In this work we introduce our event-driven vehicular mobility model (EVIMO) which considers the most important aspects of car mobility without the need of high computational power. Furthermore, we show that the structure of the road map is the dominating factor by discussing the results for different characteristic types of cities.

***Prof. Dr. Alexander von Bodisco:** E-Mail: alexander.vonbodisco@hs-kempten.de
Thomas Oberwallner: E-Mail: thomas_oberwallner@gmx.de

I Introduction

The automotive industry has shown more and more interest in wireless communication in the last couple of years due to the fact that even small embedded devices are now capable of supporting a large number of different applications. This applications range from simple monitoring to road safety, driver assistance and mobile entertainment applications. Especially, road safety and driver assistance applications opened a new field of research in computer networking since they come with high demands on the applied routing protocols in terms of stability and reliability.

As a consequence of the specific mobility pattern, establishing a self-organized network for car to car communication represents a difficult task. Routing protocols have to distinguish between opposing traffic and cars which drive into the same direction. In the following we also use the term node for a moving object. Moreover, the high relative node speed at intersections and the limited degree of freedom of the movement has to be concerned in order to establish a stable topology. The mobility constraints lead to varying node densities which have to be taken into account by the applied protocols.

Many mobility constraints and model characteristics such as movement on streets, speed limitations, movement restrictions, inter-vehicle interactions, overtaking and intersection handling affect the movement of the cars and thus the wireless communication. Other model characteristics, like the varying acceleration and size of cars, only have a minor impact on the movement.

In contrast to individual car characteristics, the road network of the simulated area has a huge impact on the mobility of cars which represents – from our point of view – the dominating factor of the movement. Therefore, the question arises to what extend mobility models for Vehicular Ad-hoc Networks (VANETs) have to consider aspects which affect the movement of cars in order to allow a meaningful simulative performance evaluation and comparison of new protocols and mechanisms. In this work we discuss the results of our optimized vehicular mobility model for different characteristic cities which considers the most important aspects of car mobility without the need of high computational power.

This paper is organized as follows. An introduction of different VANET simulators is provided in Section II. In Section III, we describe our optimized event-driven VANET mobility model. The simulated scenarios are discussed in Section IV. The simulation results are evaluated in Section V. Finally, we conclude our work in Section VI.

II Related Work

A Existing VANET simulators

In [1], 116 simulation studies in the field of Inter-Vehicle Communication (IVC) from 2009 to 2011 have been evaluated. The study compares the applied network simulator tools, the medium access protocols, the mobility model and the scenario which has been used to evaluate the simulation results. All network simulators are based on discrete-event simulation. The combined share of ns-2 [2] and ns-3 [3] of the applied network simulators is about 50% in all three years. The use of OMNeT++ [4] has increased in 2010 and 2011, when it was the network simulator with the second highest share. The share of OPNET [5] was below 10% in all three years.

Figure 1, which is adopted from [1], displays the relative share of traffic simulators which have been applied in the studies. In more than 40% of the studies, the road traffic simulator is not indicated which means that presumably no realistic movement model has been applied. SUMO has the highest share of known traffic simulators [6] with about 20% in 2009 and 2011 and about 30% in 2010. VanetMobiSim [7] is the traffic simulator with the second highest share, but its use is decreasing. VISSIM [8], a commercial simulator, has a share of about 6% within the three evaluated years. The simulators with the two highest relative shares are presented in detail in the following subsubsections. A comparison of the main features of both simulators can be found in Table I.

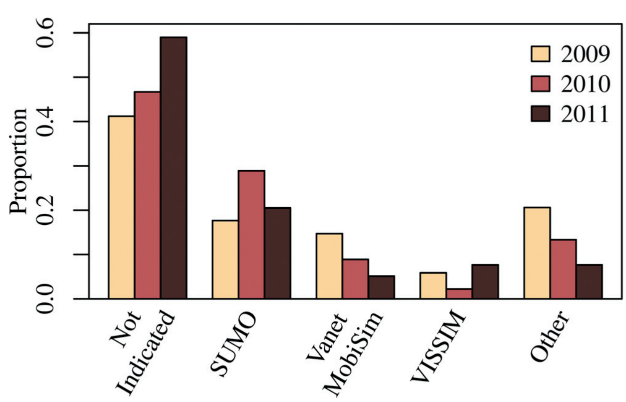


Fig. 1: Application of VANET simulators from 2009 to 2011 in studies.

The research article also compares the scenarios which have been evaluated. In about 10% of the studies, the scenario is not even indicated. About 50% of the studies use Manhattan grid as simulation area. Real world scenarios are only evaluated in about one third of the research articles.

1) Simulation of Urban Mobility (SUMO): Simulation of Urban Mobility (SUMO) is a space-continuous and time-discrete traffic simulation package [6]. It has been mainly developed by the Institute of Transportation Systems at the German Aerospace Centre since 2000, written in C++ and released under the GPL. SUMO can be executed under Windows as well as under Linux. The behaviour of drivers is modelled microscopically. Its main features include a collision free vehicle movement with different vehicle types, single vehicle routing, multi-lane streets with lane changing and dynamic routing. Besides, right of way rules and traffic signals are possible and an OpenGL graphical user interface is included. The driver model of SUMO is described in detail in [9]. SUMO's default driver model is the car following model by Krauss [10] with some modifications.

2) VanetMobiSim: VanetMobiSim is a microscopic and macroscopic vehicular traffic simulator [7]. It is written in Java and is an extension to the CANU Mobility Simulation Environment (CanuMobiSim) [11] which is able to import geographical data files and offers some synthetic mobility and vehicular mobility models. Among others, CanuMobiSim implements Random Waypoint, Brownian Walk and GaussMarkov Walk. The current stable version of VanetMobiSim (1.1) has been released in February 2007 and is able to import US Census Bureau (TIGER/Line) maps [12]. The import of OpenStreetMap data has been introduced in the latest beta version (2.0.1). All nodes are instances of mobility models and after every step of time the act-method of every node is called. According to the mobility model, the nodes then adjust their position and the movement vector is updated. The mobility model is mainly vector-based, but direct manipulations of positions are possible, too. This feature is used, when a car would exceed its destination within the next time step. Then the new position is set to the destination. In addition to the synthetic models mentioned above, CanuMobiSim implements the following mobility models:

- Constant Speed Motion (CSM)
- Smooth Motion Model (SMM) [13]
- Fluid Traffic Model (FTM) [14]

VanetMobiSim introduces new movement models which behave similar to normal cars. The motion models are (in contrast to some at CanuMobiSim) independent of the trip of the cars. Thus, the trip generation and the movement of the cars are separated. VanetMobiSim offers the following mobility models:

- Intelligent Driver Motion (IDM) [15]
- Intelligent Driver Motion with Intersubsection Management (IDM-IM) [7]
- Intelligent Driver Motion with Lane Changing (IDM-LC) [7]

The functionality of Sumo and VanetMobiSim is summarized in Table I which is based on information provided in the survey of Martinez et al. [16].

Table 1: Comparison of existing VANET simulators

	SUMO	VanetMobiSim
Import of OSM-files	Yes, with NETCONVERT	Not in stable version, only in beta
Mobility model	Krauss (modified), IDM	IDM, IDM-IM, IDM-LC
Trip generation	according to flow definitions,randomly, OD matrices or population statistics	random, sightseeing
Routing	A* (fast)	Dijkstra (slow)
Statistics	Node density	Position dump, edge lane traffic, trip/route information, not aggregated
Import/Export traces	No/Yes (ns-2, GloMoSim, QualNet, NET)	No/Yes (unknown format)
Network simulation	No	No

III Event-driven mobility modeling

In general, there are different ways to implement a mobility model. One approach is to implement a function which updates the speed and direction of a car. The function has to be called periodically for every car. This procedure has been selected by the developers of SUMO and CanuMobiSim/VanetMobiSim. Since there are no acceleration and no reaction times considered, we followed a different approach for the EVIMO model. Instead, cars schedule their own events, when action has to be taken. If a following car might have to react, cars will schedule events to their proceeding car. Thus, the events correspond to actions of cars. The advantage of scheduling events only when necessary is that the number of interrupts is very small when the number of cars is small, since cars do only have to change their direction or speed when arriving at nodes. When increasing the number of cars, the number of events also increases, since the speed might have to be adapted between nodes.

Node arrival

The main interrupt is the node arrival interrupt. A flowchart of this interrupt is displayed in Figure 2. The node arrival interrupt is triggered when a car arrives at a node and is the only event when a car changes its direction. Additionally, the position of the car is logged. If a route is imported, this will be the only interrupt which is scheduled and the new ground speed and direction is calculated based on the imported route. At the next imported timestamp, a new node arrival event will be scheduled.

If the route is not imported, it will be checked whether there is a car in front. If there is no car in front, the new ground speed will be set depending on the speed limit of the road section and the desired speed of the car. If the next node is an intersection an intersection arrival event will be scheduled, otherwise a node arrival event will be scheduled. If there is a car in front, the new ground speed of the following car will depend on the speed and distance to the car in front. If the car in front is driving slower, it will be checked whether the following car approaches to the minimum distance before the next node arrival. If the minimum distance will be reached, before the next node arrival, a brake in front event will be scheduled for the following car when the minimum distance will be reached. Otherwise, the following car will drive its desired speed and a node arrival respectively intersection arrival event will be scheduled.

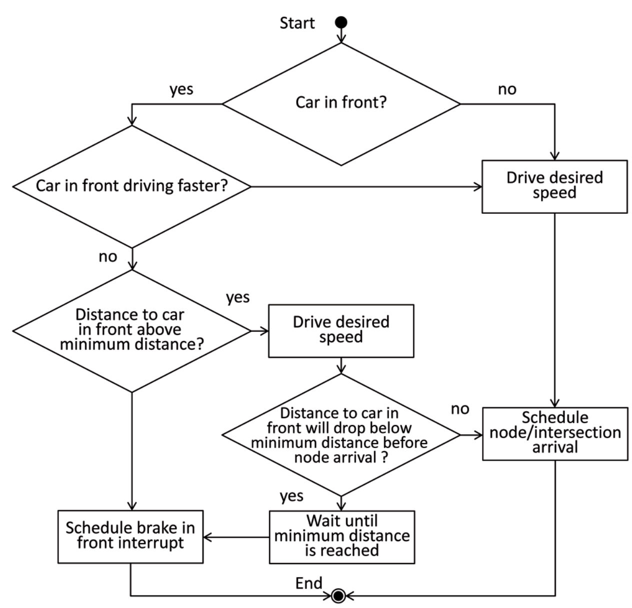


Fig. 2: Flowchart of Node Arrival Interrupt.

Intersection arrival

Intersection arrivals are triggered when cars arrive at intersections. A flowchart of the interrupt is displayed in Figure 3. This interrupt checks whether a car is able to pass the intersection. In case the passing is possible, a node arrival event is scheduled, otherwise the car will stop and another intersection arrival will be scheduled.

If there is no traffic signal, the crossings of the intersection within the previous 2 seconds will be checked for an incompatible crossing. If there was no incompatible crossing, it will be checked whether the car can enter the next section (the distance to the car in front has to be more than 2 meters). If entering the section is possible, a node arrival event will be scheduled and the car is entered in the list of successful crossings, otherwise the car will stop and an intersection arrival event bwill be scheduled 1 second later. If there was at least one incompatible crossing within the last 2 seconds, the car will stop and an intersection arrival event will be scheduled 2 seconds after the last successful crossing of the intersection.

If there is a traffic signal at the intersection, it will be checked whether the signal is green in the direction in which the car arrives. If it is green, it will be checked whether the car can enter the next section (the distance to the car in front has to be more than 2 meters). If the car can enter the next section, a node arrival event will be scheduled, otherwise the car will stop and an intersection arrival event is scheduled 1 second later. If the traffic light is red, the car will stop.

Every time the car stops, it is checked, whether there is a car behind. If there is a car behind, a brake in front interrupt will be scheduled to the car behind.

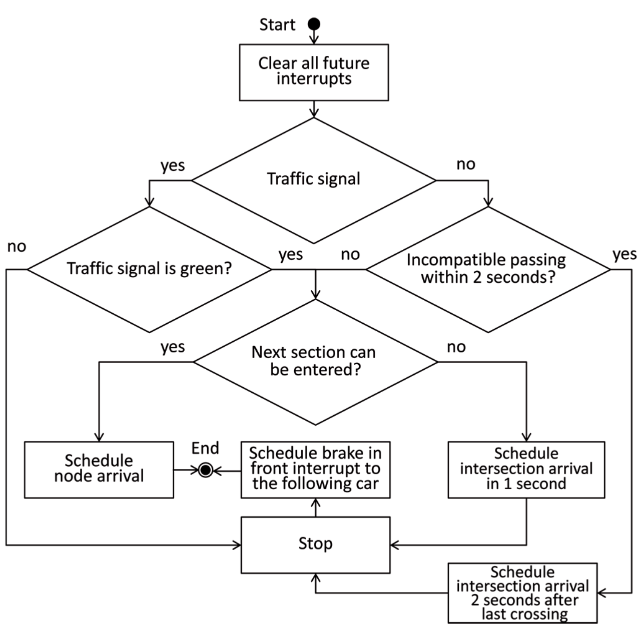


Fig. 3: Flowchart of Intersection Arrival Interrupt.

Brake in front

The brake in front interrupt is the only interrupt which can change the ground speed of a car when not being at a node. It is scheduled by the car in front when decelerating or by itself when the minimum distance to the car in front cannot be guaranteed. The flowchart of this interrupt is displayed in Figure 4. This interrupt checks whether there is a car in front and sets the new ground speed of the following car depending on the distance to the car in front and speed of the car in front.

If the distance to the car in front is exceeding the minimum distance, the following car will drive its desired speed and a new brake in front interrupt will be scheduled if the distance to the car in front will drop below the minimum distance before the next node arrival. Otherwise a node arrival or intersection arrival event will be scheduled depending on the type of the node. If the distance to the car in front is equal to the minimum distance, the following car will drive the speed of the car in front (if it is smaller than the desired speed of the following car) and a node arrival or intersection arrival event will be scheduled. If the following car under runs the minimum distance, its own ground speed will be set 5% lower than the ground speed of the car in front. If this is sufficient to increase the distance a brake in front interrupt is scheduled. Otherwise, a node arrival will be scheduled. Every time a car changes its ground speed while processing a brake in front event, a brake in front interrupt is scheduled to the car following the current car which then itself adjusts its ground speed.

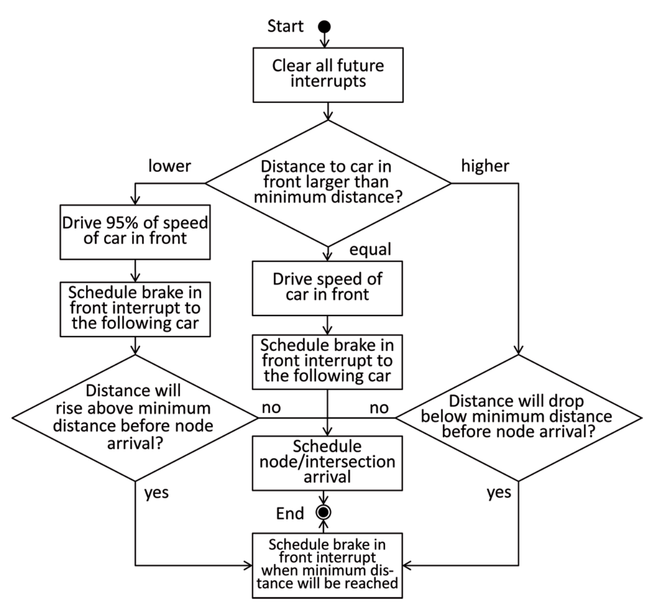


Fig. 4: Flowchart of Brake In Front Interrupt.

Pause completion

The pause completion interrupt processes the departure of paused cars. It checks whether there is a car within 50 meters in front and 50 meters behind the resuming car. If there is no car, a node arrival event will be scheduled, otherwise another pause completion event will be scheduled 1 second later.

Overtake completion

The overtake completion event processes completed overtaking manoeuvres. It is scheduled when the car which is overtaking is on the same position on the road as the car which is being overtaken. In this event, the position of the car which is overtaking and the car which has been overtaken is switched in the list of cars on the current section. Additionally, a brake in front interrupt for the car which is being overtaken is scheduled.

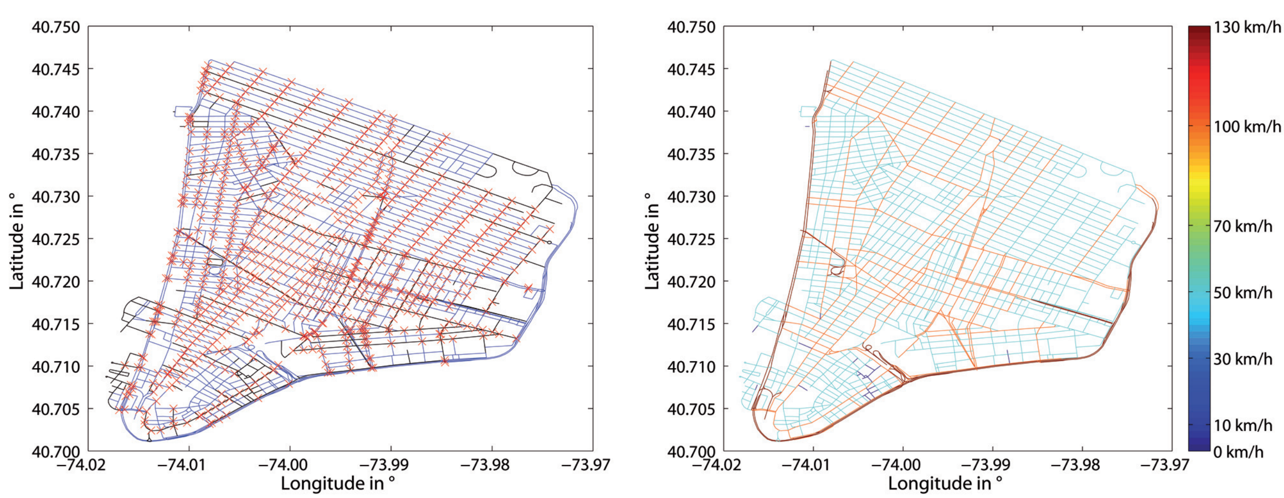
IV Introduction of scenarios

In reality, cities and their traffic infrastructures differ strongly. Therefore, three scenarios with very characteristic properties have been chosen:

– Lower Manhattan (Figure 5(a)): The island of Manhattan is an interesting map since there is a characteristic design with many parallel roads which are leading from north to south and east to west. Since the map of the whole island would be too large, only Lower Manhattan with a rectangle size of 19.72 km2 has been cut out. The boundaries in the east, west and south are clearly defined (with the East River and the Hudson River). In the north, the 20th street has been chosen, since there is no access to the motorway in the east from 14th street and this part of the motorway could therefore not be accessed. The map of Lower Manhattan has 1309 intersections and 752 traffic signals.
– Soest (Figure 5(c)): This map has unique characteristics. The city is surrounded by radial concentric ring roads. Since the city has no natural boundaries and to avoid influences of suburban areas, the map has been cut on the outermost ring road. Soest has 925 intersections and 17 traffic signals.
– Regensburg (Figure 5(e)): The city of Regensburg was chosen as a historic city which has not been planned in any specific manner as the other two scenarios presented above. Unlike Soest, it contains motorways. Besides, its boundaries are given by the Danube in the north and west, the motorway A3 in the south and the Osttangente in the east. The map of Regensburg contains 1554 intersections and 165 traffic signals.

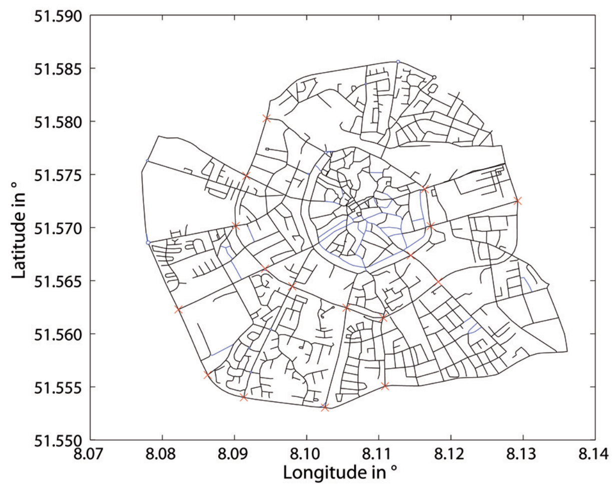
Table 2: Dimensions and total length of the road network

City	Dimensions of scenario (North-South x East-West)	Total length of road network
L. Manhattan	5000.48 m x 3943.04 m = 19.72 km2	302,968 m
Soest	3647.58 m x 4083.52 m = 14.89 km2	266,154 m
Regensburg	4532.84 m x 9200.45 m = 41.70 km2	502,971 m

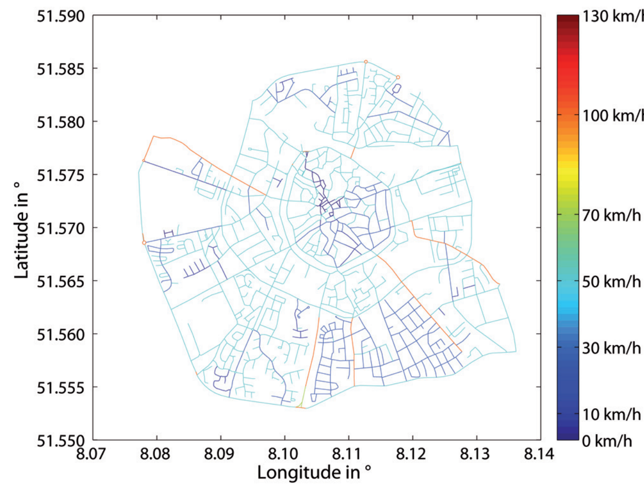


(a) Road network of Lower Manhattan

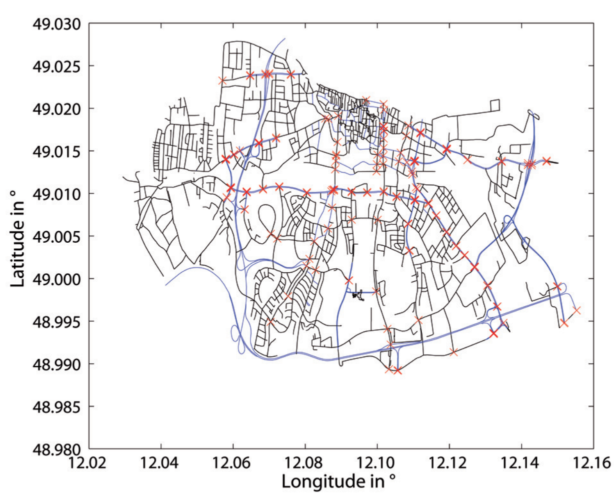
(b) Speed limits in Manhattan



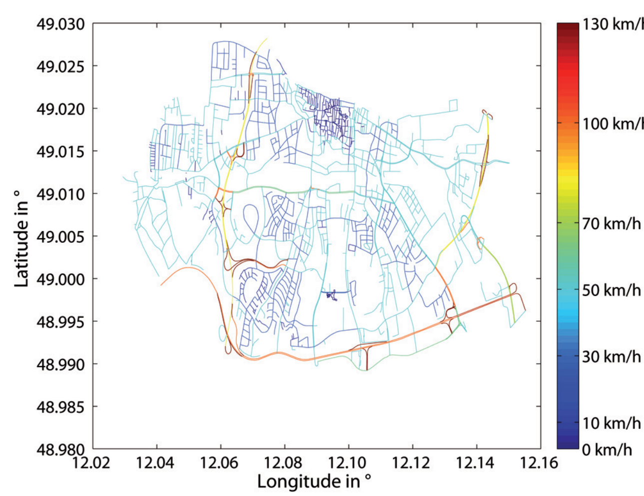
(c) Road network of Soest



(d) Speed limits in Soest



(e) Road network of Regensburg



(f) Speed limits in Regensburg

Fig. 5: Road networks and speed limits of the scenarios.

A Types of roads

Table III displays the types of roads in the scenarios. The highway type with the highest relative ratio is residential, which are roads in residential areas, in all scenarios. The highway types with the next highest relative share in Lower Manhattan are primary (~15.5%) and secondary (~14%), in Soest they are secondary (~13%) and tertiary (~7%) and in Regensburg they are tertiary (~10%) and living street (~6%). The map of Soest does not include motorways/trunks or their respective links while the road networks of Lower Manhattan and Regensburg include about 10% and 9% motorways/trunks. In Lower Manhattan, about 20% of the highways are not classified whereas in the other maps only about 2% are unclassified. Pedestrian roads have not been filtered but cannot be accessed by cars. Tracks in Lower Manhattan have been filtered since they often were not connected to the road network.

Table IV displays the number of ways which are tagged as one-way and the length of them in the scenarios. Since motorways are always one-way streets and Soest does not contain motorways, Soest has the least one-way streets. Most of them are located in the southern part of the city-centre. Thus, they have the least mean distance with 131.7 meters per one-way street. In Regensburg, mostly motorways and some ring roads are one-way streets. Their average distance is 229.4 meters. Lower Manhattan has the most one-way streets and many ways are not subdivided at crossroads, therefore the average distance is the largest with 319 meters.

Table 3: Types of roads in the scenarios (absolute length)

	absolute length [meters]		
	Manh.	**Soest**	**Reg.**
Motorway	15,256	0	22,087
Motorway link	6,268	0	14,959
Trunk	9,221	0	4,989
Trunk link	429	0	2,107
Primary	46,885	10,557	16,554
Primary link	64	0	2,406
Secondary	43,373	34,350	0
Secondary link	177	92	0
Tertiary	3,104	18,264	48,879
Tertiary link	0	0	0
Living street	0	9,831	29,082
Pedestrian	7,744	5,562	26,549
Residential	110,061	177,921	301,083
Unclassified	60,387	3,617	10,352
Track	0	5,961	23,925
Total length	302,968	266,154	502,971

Table 4: One-way streets in the scenarios

City	Number of one-way streets	Total length of one-way streets	Average length of one-way streets
L. Manhattan	627	200,228 m	319.34 m
Soest	64	8,427 m	131.67 m
Regensburg	440	100,915 m	229.35 m

B Speed limits

Figure 6 contains the relative share of speed limits in the maps before (blue) and after assigning speed limits to the roads which have no data in the map file. In Lower Manhattan, no speed limits are contained in the map data (when ignoring pedestrian roads). The share of roads with no speed limit is 97%. After assigning speed limits to ways with an unknown speed limit according to the default speed limit of the type of highway and the presence of residential areas within the roads, almost all motorways get a speed limit of 130 km/h (~10%). Residential roads and highways which cross residential areas get a speed limit of 50 km/h (~60%). All remaining roads have a speed limit of 100 km/h. In Soest, only about half of the ways (~57%) have no speed limit set in the map data. Speed limits other than than 0, 10, 50, 100 or 130 km/h are present in one quarter (~23%) of the ways, which mainly consists of ways with a speed limit of 30 km/h. After assigning speed limits to all roads, most of the roads (~66%) have a speed limit of 50 km/h. Since there are no motorways, no road has a speed limit of 130 km/h. Regensburg has the lowest share of roads with no speed limit in the map data (~48%) but the highest share of roads with a speed limit set to other values (~33%). Similar to Soest, most of the roads with a speed limit set to values different from 0, 10, 50, 100 or 130 km/h have a speed limit of 30 km/h. After assigning speed limits to the remaining roads, about half of the roads (~50%) have a speed limit of 50 km/h.

The speed limits of the cities are displayed in Figures 5 (b), 5(d) and 5(f). The color of the roads corresponds to the speed limit on the road. In Lower Manhattan, most of the inner roads seem to get a correct speed limit of 50 km/h, but some roads get assigned a speed limit of 100 km/h, which should have a speed limit of 50 km/h, too. The belt highways are assigned a speed limit of 130 km/h. In Soest, most of the speed limits seem to be correct as well, since only few roads have a speed limit of 100 km/h. The dark blue areas are pedestrian streets, living streets or roads with a speed limit imported from the map data. The speed limits in Regensburg also seem to be valid. All areas with a speed limit below 50 km/h have a correct speed limit since they are assigned based on speed limits in the map file or highway types with a fixed speed limit. Higher speed limits are only assigned to the motorways and one ring road in the east of Regensburg.

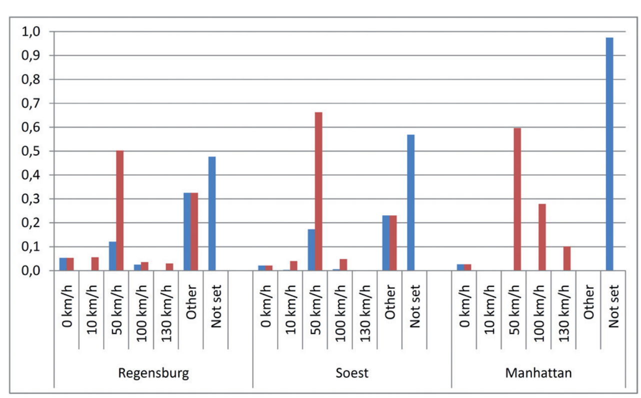


Fig. 6: Share of speed limits in map data (blue) and after guessing unknown speed limits (red).

C Distance between nodes/intersections

Figure 7 displays the distances between successive nodes in ways and Figure 8 shows the distance of ways between two consecutive intersections. In Lower Manhattan, the mean distance between intersections is 105.4 meters with a standard deviation of 95.1 meters and a median of 80.0 meters. The 10% quantile is 35.0 meters and the 90% quantile is 181.1 meters. The distances between nodes are shorter, which is quite obvious since nodes represent the course of the road. Their mean is 64.3 meters with a standard deviation of 61.0 meters and a median of 56.5 meters. The 10% quantile is 7.0 meters and the 90% quantile is 138.1 meters. About 35% of the distances between intersections are between 70 and 90 meters. About 23% of the nodes have a distance between 69 and 90 meters. Both peaks reflect the typical distance within the grid of parallel and straight roads. Other local peaks at about 160 meters and 240 meters are multiples of the normal grid distance. The median of the distances between nodes being smaller than the grid distance and the distribution of the distances is also normal since multiple nodes are needed to model bends in the roads. The distances of nodes and intersections in Soest and Regensburg do not show distinctive peaks like in Lower Manhattan since the road network has not been created with parallel roads and equal distances between these roads.

In Soest, the mean distance between intersections is 112.6 meters with a standard deviation of 95.6 meters and a median of 87.9 meters. The 10% quantile is 29.8 meters and the 90% quantile is 219.1 meters. The mean distance between nodes is 51.9 meters with a standard deviation of 45.2 meters and a median of 40.3 meters. The 10% quantile is 8.86 meters and the 90% quantile is 107.0 meters.

In Regensburg, the mean distance between intersections is 129.0 meters with a standard deviation of 159.3 meters and a median of 89.2 meters. The 10% quantile is 22.1 meters and the 90% quantile is 262.9 meters. The mean of distances between nodes is 36.6 meters with a standard deviation of 41.8

meters and a median of 24.2 meters. The 10% quantile is 3.64 meters and the 90% quantile is 83.9 meters. The distribution of the distances between nodes has its peak at distances between 0 and 3 meters which is caused by pedestrian paths which are modelled with many nodes and do not reflect the course of the roads which cars can drive on. The quantiles of distances between nodes and intersections are summarized in Table V.

Table 5: Quantiles of distances between nodes and intersections in the scenarios

		Manhattan	Soest	Regensburg
Intersec-tions	Mean:	105.41 m	112.58 m	128.99 m
	Stddev:	95.06 m	95.63 m	159.30 m
	Median:	80.00 m	87.93 m	89.24 m
	10% quantile	34.97 m	29.80 m	22.07 m
	90% quantile	181.05 m	219.10 m	262.90 m
Nodes	Mean:	64.30 m	51.91 m	36.59 m
	Stddev:	60.98 m	45.16 m	41.76 m
	Median:	56.51 m	40.31 m	24.16 m
	10% quantile	7.04 m	8.85 m	3.64 m
	90% quantile	138.14 m	107.03 m	83.86 m

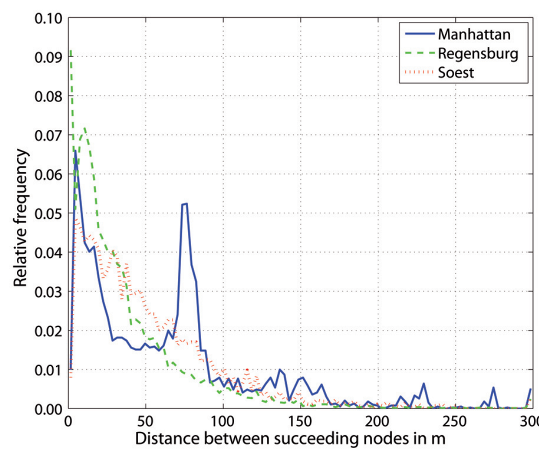


Fig. 7: Distance between nodes (PDF).

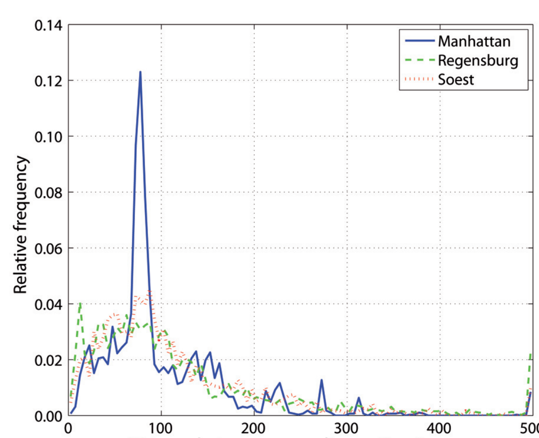


Fig. 8: Distance between vertices (PDF).

V Evaluation

We simulated three cities with different road networks in order to evaluate the impact on the movement. As outlined in the previous section, the lengths of the corresponding road maps differ. Therefore we had to simulate a different number of cars for each scenario to create a comparable environment. The Lower Manhattan scenario is selected as the reference scenario in which we simulate 500, 1000 and 1500 cars. In this work we focus on scenarios with relatively low car density due to the fact that the impact of the road network on the movement can be evaluated without unwanted side effects. The number of cars in the other scenarios are scaled according to the lengths of their road maps such that the node density is equal in all scenarios. Thus, we simulate 807, 1604 and 2421 cars in the Regensburg scenario while the number of cars in the Soest scenario are set to 441, 883 and 1324 cars. All cars choose random intersections as destinations. The distribution of the destinations is not dependent on the number of cars and therefore is identical for all configurations. The simulation duration is set to 12 hours. Every traffic signal gets a uniform offset between 0 seconds and 20 seconds and a uniform duration between 5 seconds and 15 seconds in order to avoid synchronous signal switching. The cars are evenly distributed at the beginning and start to move after 60 seconds.

Mobility can be characterized by many different parameters, e.g. the node distribution, absolute speed distribution, relative speed distribution, number of neighbor nodes, average distance between nodes and many many more. In this work, we focus on the absolute and relative node speed, as well as the average number of neighbors due to the fact that they represent the most important factors for wireless communication and are used in the majority of mobility surveys.

As outlined in Table VI, the average speed of cars in the Lower Manhattan scenario decreases from 60.5 km/h (500 cars) to 56.68 km/h (1000 cars) and 52.13 km/h (1500 cars) for the scenario with the highest node density. About 40% of the cars have to stop temporarily when simulating 1500 cars, whereas about 35% have to stop when simulating 500 cars (Figure 9).

Table 6: Average speed for different numbers of cars

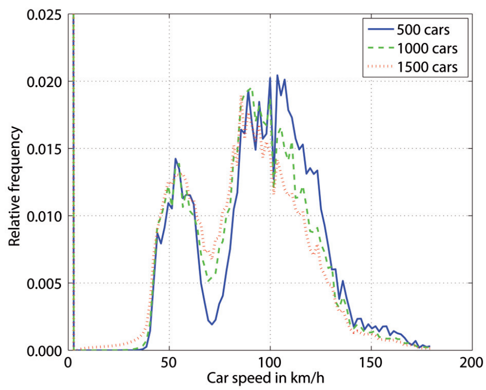
Scenario	Number of cars	Average speed [km/h]
Lower Manhatten	500	60.50
Lower Manhatten	1,000	56.68
Lower Manhatten	1,500	52.13
Regensburg	807	48.87
Regensburg	1,614	31.02
Regensburg	2,421	21.17
Soest	441	48.87
Soest	883	23.86
Soest	1,324	15.88

In the Regensburg scenario, the average speed is dropping from 48.87 km/h to 31.02 km/h and 21.17 km/h when raising the number of cars (Table VI, Figures 10). The impact that the node density has on the average node speed is much higher in the Regensburg scenario compared to the Manhattan scenario (60.50 km/h at 500 cars vs. 52.13 km/h at 1500 cars). The number of waiting cars rises from 10% to 45%. As a consequence, many cars are waiting behind another car or at an intersection. With a lower average speed, the average of the relative speed decreases, too.

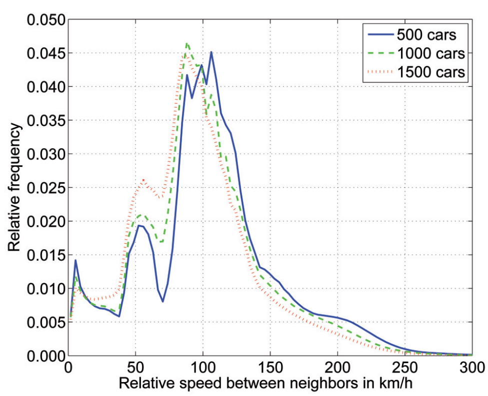
The traffic in the Soest scenario is most sensitive to changes in the node density as indicated in Table VI and Figure 11. The average node speed again is decreasing when the number of cars is increased. The average speed is 48.87 km/h, when simulating 441 cars, 23.86 km/h when simulating 883 cars and 15.88 km/h when simulating 1324 cars (Table VI). With 441 cars, less than 10% of the cars are standing still at the same time, while more than 60% of the cars are standing still when simulating 1324 cars.

The robustness of a wireless multihop network is mainly affected by the relative speed between neighbors which has a direct impact on the link duration. Link duration represents the time during which two mobile nodes may communicate with each other before the connection is lost due to the movement. The probability density function of the relative speed between neighbors in the Manhattan scenario is shown in Figure 10(b). The slope of the curve shows three characteristic peaks at 0 km/h, 50 km/h and 100 km/h. The first two peaks are caused by cars waiting at an intersection while other cars pass by. Cars moving on highways which go through the city center are responsible for the long tail of the PDF. With increasing number of cars, the PDF is shifted towards the lower speeds and the peak at 50 km/h is further increasing.

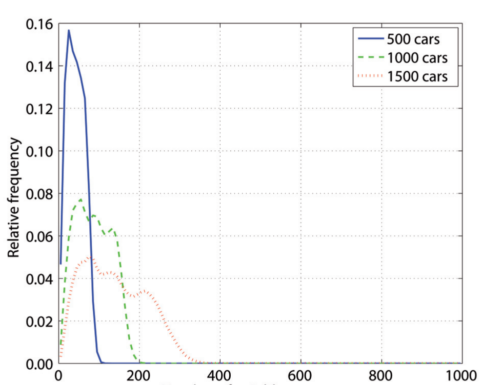
The PDF of the number of neighbors in the Manhattan scenario is presented in Figure 9(c). The number of neighbors is increasing with higher node density, as expected. The figure indicates that the average number of neighbors for the 1000 cars scenario is approximately twice is high as in the 500 cars scenario. The number of neighbors in the 1500 cars scenario is about four times as high as in the 500 cars scenario. The huge increase is mainly caused by cars waiting at intersections. Due to the fact that the waiting time at intersection increases, waiting cars are ideal candidates for forwarding data traffic, especially if they have to wait for more than one traffic light cycle to pass the intersection.



(a) Car speed PDF



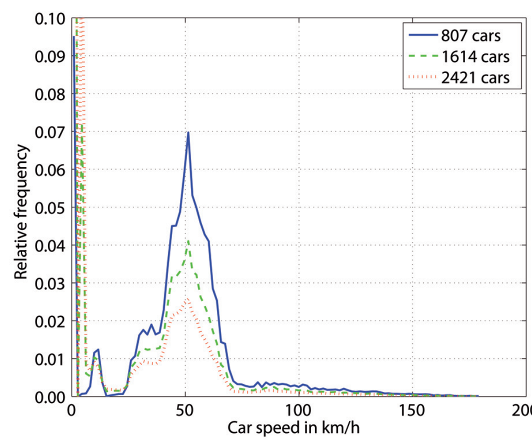
(b) Relative speed between neighbors PDF



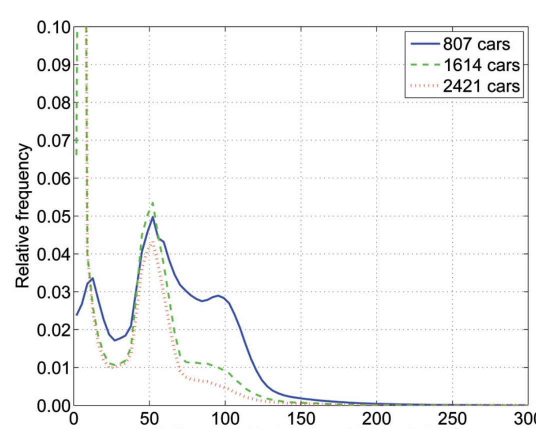
(c) Number of neighbors per car PDF

Fig. 9: Mobility metrics in Lower Manhattan for different numbers of cars.

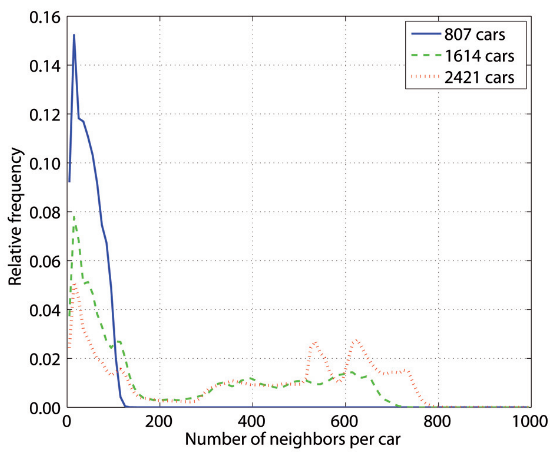
The slope of the PDF of the car speed for the Regensburg scenario (c.f. Figure 10(a)) has a small peak close to 0 km/h and one characteristic peak at 50 km/h. Both peaks are again caused by waiting cars. The peak is higher compared to that in the Manhattan scenario since there is only one highway in the Regensburg scenario. Thus, the majority of roads have a speed limit of 50 km/h. For the same reason, no significant peak at 100 km/h can be recognized. The PDF of the relative speed between neighbors is shown in Figure 10(b). The slope is similar to that in the Manhattan scenario, but it is slightly smoother as a consequence of the smaller fraction of highways in the road network. In addition, the figure reveals that the number of cars have a higher impact on the relative speed than in the Manhattan scenario. The reason for this lies in the structure of the road network. The map of Manhattan is regular, structured and represents an almost perfect grid.



(a) Car speed PDF

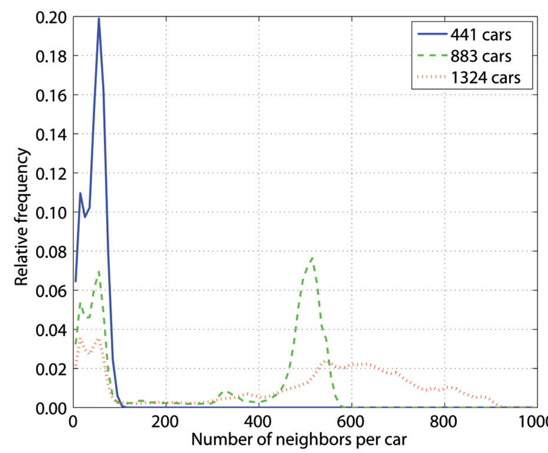


(b) Relative speed between neighbors PDF



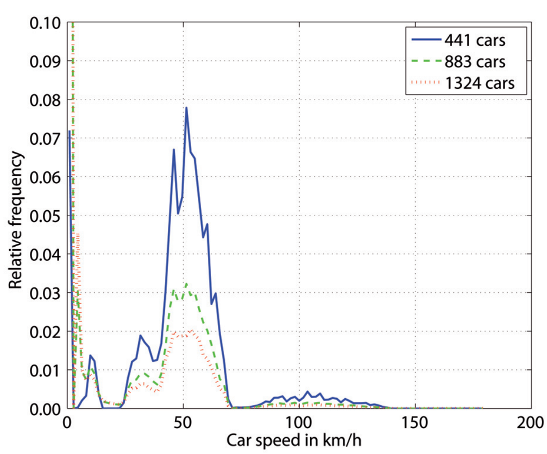
(c) Number of neighbors per car PDF

Fig. 10: Mobility metrics in Regensburg for different numbers of cars.

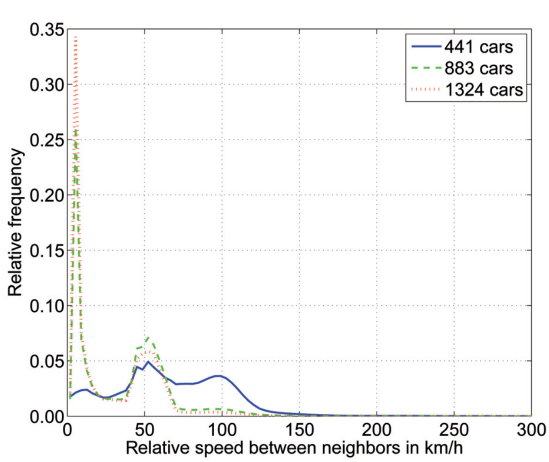


(c) Number of neighbors per car PDF

Fig. 11: Mobility metrics in Soest for different numbers of cars.



(a) Car speed PDF



(b) Relative speed between neighbors PDF

Thus, many shortest routes between origin and destination exist which results in a higher flexibility and a more evenly distributed node density. Therefore, the traffic in Manhattan is less affected by the node density than the traffic in Regensburg.

As outlined above, the number of neigbors in a VANET is crucial for establishing a stable network. The evaluation of the Manhattan scenario has shown that average number of neighbors doubled if the number of cars is increased by 500. The PDF of the number of neighbors in the Regensburg scenario, which is shown in Figure 10(c), points out that the traffic in this scenario is affected in another way due to the huge differences in the road network. The scenario with the smallest number of cars has a typical slope. Most of the cars have a small number of neighbors and almost no car has more than 160 neighbors. This indicates that cars are evenly distributed on the road network. However, if the node density is increased a significant change in the PDF can be recognized. The new PDF has a bimodal characteristic. There is still a large fraction of nodes which have a small number of neighbors. Besides this group, a second group of nodes can be identified which have a rather large number of neighbors. The reason for this bimodal shape of the PDF is that some nodes have to stop at intersections and/or are stuck in a traffic jam while other nodes are able to move more or less without any interruption to their target destination. This latter fraction of nodes are responsible for the probability peak between 0 and 150 neighbors.

The road network of Soest has a completely different structure compared to Manhattan and Regensburg. Therefore one would assume a different distribution of the car speed and number of neighbors. A first look at the car speed PDF shown in Figure 11(a) does not confirm this assumption since the slope of the PDF looks similar to the car speed PDF in the Regensburg scenario. Thus, the traffic flows are comparable to the flows in the previous scenario. However, a big difference can be recognized when comparing the PDFs of the relative speed between neighbors. Figure 11(b) shows that a large fraction of neighbors have a low relative node speed which is caused by the concentric topology of the road network of Soest. This effect becomes dominating as soon as nodes start to queue at intersections. The PDF of the number of neighbors (c.f. Figure 11 (c)) shows a bimodal shape as a consequence of the queueing effect at intersections whereas the second peak is more dominant for the scenario with medium node density.

VI Conclusion

The question what is realistic or typical movement in VANETs cannot be answered since the behavior of drivers varies and is strongly influenced by the underlying road network. Moreover, traffic in real-world scenarios changes over time similar to a computer network. During the rush hour the characteristics such as number of neighbors and link duration are different compared to those during the rest of the day.

Our comparison of different cities revealed that the road network and the node density are two of the most dominating factors. In addition, the results have shown that every city has its characteristic absolute and relative speed distribution. The distribution of the number of neighbors has a bimodal shape for scenarios with medium and high node density where the underlying road map does not follow a grid structure. Our future work will include a detailed evaluation of a large selection of popular cities to provide other researchers valuable input for their simulations.

References

1 S. Joerer, C. Sommer, and F. Dressler, *Toward Reproducibility and Comparability of IVC Simulation Studies: A Literature Survey*, IEEE Communications Magazine 50(10): 82–88, October 2012.
2 NS-2: Network Simulator. http://nsnam.isi.edu/nsnam/. Retrieved: 2013-05-08.
3 NS-3: Network Simulator. http://www.nsnam.org/wiki/. Retrieved: 2013-05-08.
4 OMNeT++: Network Simulator. http://www.omnetpp.org/. Retrieved: 2013-05-08.
5 OPNET-Technologies. OPNET modeler university program. http://www.opnet.com/services/university/. Retrieved: 2013-05-08.
6 M. Behrisch, L. Bieker, J. Erdmann and D. Krajzewicz. *SUMO-Simulation of Urban MObility-an Overview*, In SIMUL 2011, The Third International Conference on Advances in System Simulation, pages 55–60, 2011.
7 J. Harri, M. Fiore, F. Filali and C. Bonnet. *Vehicular mobility simulation with VanetMobiSim*, Simulation 87(4): 275–300, 2011.
8 PTV Vissim traffc simulation. http://vision-traffc.ptvgroup.com/de/produkte/ptv-vissim/. Retrieved: 2013-05-08.
9 D. Krajzewicz. *Traffc Simulation with SUMO – Simulation of Urban Mobility*. Fundamentals of Traffc Simulation, International Series in Operations Research & Management Science, Springer, pages 269–293, 2010.
10 S. Krauss, P. Wagner and C. Gawron. *Metastable states in a microscopic model of traffic flow*. Physical Review E 55(5): 5597–5602, 1997.
11 I. Stepanov, J. Hahner, C. Becker, J. Tian and K. Rothermel. *A metamodel and framework for user mobility in mobile networks.*, In Proc. of 11th IEEE International Conference on Networks (ICON 2003), pages 231–238, 2003.
12 United States Census Bureau. Topologically Integrated Geographic Encoding and Referencing Products. https://www.census.gov/geo/mapsdata/data/tiger.html. Retrieved: 2013-02-09.
13 C. Bettstetter. *Smooth is better than sharp: a random mobility model for simulation of wireless networks*. In Proceedings of the 4th ACM International Wokshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems. ACM, pages 19–27, 2001.
14 I. Seskar, S.V. Maric, J. Holtzman and J. Wasserman. *Rate of location area updates in cellular systems*. In Proc. of 42nd IEEE Vehicular Technology Conference, pages 694–697, 1992.
15 M. Treiber, A. Hennecke and D. Helbing. *Congested traffc states in empirical observations and microscopic simulations*. Physical Review E 62(2): 1805–1824, 2000.
16 F.J. Martinez, C. Toh, J. Cano, C.M.T. Calafate and P. Manzoni. *A survey and comparative study of simulators for vehicular ad hoc networks (VANETs)*. Wireless Communications and Mobile Computing 11(7): 813–828, 2011.

Prof. Dr. Alexander von Bodisco: Technology Centre for Wireless Communication, University of Applied Sciences Kempten

Thomas Oberwallner: Chair for Network Architectures and Services, Technical University Munich