

ASH: Application-aware SWANS with Highway mobility

Khaled Ibrahim and Michele C. Weigle

Department of Computer Science

Old Dominion University

Norfolk, VA 23529-0162

{*ibrah_k, mweigle*}@cs.odu.edu

Abstract—Most VANET simulators do not allow for feedback between the vehicle mobility model and the network simulator. This limits the realistic simulation of safety and traffic information applications that might cause drivers to change their routes. To address this issue, we developed extensions to the SWANS wireless network simulator. Our SWANS modules, which we collectively call ASH (Application-aware SWANS with Highway mobility), make several contributions. ASH allows for the needed two-way communication between the mobility model and the networking model. To support highway scenarios, we built in ASH a customizable highway topology, allowing entrances, exits, and variable number of lanes. ASH includes implementations of the existing IDM car-following and MOBIL lane-changing models, as well as the Inter-Vehicle Geocast data dissemination protocol. To facilitate diverse simulations, ASH enhances the node model in SWANS to allow for the presence of non-participating vehicles and obstacles, along with participating vehicles and road-side infrastructure. These additions to the scalable SWANS simulator will allow for realistic VANET simulations of important safety and traffic information applications.

I. INTRODUCTION

The simulation of Vehicular Ad-hoc Network (VANET) applications not only requires simulating the wireless communication between the vehicles, but also requires simulating the mobility of the vehicles. Unfortunately, these two aspects of VANET simulation have been decoupled. Both vehicular mobility and wireless communication have large communities concerned with their modelling and simulation, so high quality simulators exist in each of these areas. The problem is in merging the two types of simulators. An ideal VANET simulator would consist of two sub-simulators, a network simulator to simulate the wireless communication between the vehicles and a traffic simulator to simulate the vehicles' mobility. VANET applications that run at the top level of the network simulator can then be categorized according to how these two sub-simulators need to communicate.

Infotainment-related applications, including Internet connectivity [1], multimedia applications, and peer-to-peer applications [2], [3], require only *one-way communication* between the two sub-simulators, from the traffic simulator to the network simulator. The network simulator uses the provided information (position information, speed, acceleration, direction, *etc.*) from the traffic simulator for data routing. Since this kind of application has no effect on driving decisions (*i.e.* vehicle mobility), there is no need to generate the mobility

information dynamically. For simplicity, a pre-generated trace file of the vehicles' mobility is often used.

Safety-related and traffic information applications require *two-way communication* between the traffic simulator and network simulator. In these types of applications, the traffic simulator feeds the network simulator with position information, speed, acceleration, direction, *etc.* The VANET application that runs at the top level of the network simulator incorporates this information with surrounding vehicles' information in order to notify the driver of upcoming congestion or a possible collision. Based on this notification, driving decisions (*i.e.* vehicle mobility) may be affected. For example, the driver may choose to change lanes. These mobility decisions need to propagate back to the traffic simulator to be reflected in the vehicle mobility information. In this way, the mobility model becomes *application-aware*.

Simulating one-way communication is straightforward and is already supported by the combination of various traffic mobility and network simulators. On the other hand, the difficulty in simulating two-way communication resides in how to couple independent traffic simulators with network simulators. Recently, many VANET-specific simulators have been developed, but most allow only one-way communication.

We present extensions to the Scalable Wireless Ad hoc Network Simulator (SWANS) [4]. These extensions, which we call ASH (Application-aware SWANS with Highway mobility), consist of the following additions:

- Methods to provide feedback between the application layer and the mobility model, allowing for the needed two-way communication
- A customizable highway topology, supporting multiple lanes, entrances, and exits
- An implementation of the Intelligent Driver Model (IDM) [5] car-following model
- An implementation of the MOBIL [6] lane change model
- An implementation of the Inter-Vehicle Geocast (IVG) [7] broadcasting technique, as well as an implementation of the probabilistic IVG (p-IVG) [8] enhancement to IVG
- An enhanced node model allowing for the addition of non-communicating vehicles, road-side units, and obstacles to a simulation
- Statistical and logging facilities for reporting simulation metrics

II. RELATED WORK

Without a doubt, the existence of a standard VANET simulator that the entire research community trusts would enhance the research quality and shorten the development cycle of any VANET application. Towards this goal, much effort has been exerted to develop a VANET-specific simulator by integrating a network simulator with a mobility generator.

There have been several efforts to combine network and mobility simulators to achieve one-way communication. Karnadi *et al.* [9] developed a tool that interfaces with the mobility generator SUMO [10] to generate vehicle mobility traces that can be imported into ns-2 [11] or QualNet [12]. CanuMobiSim [13] is a mobility simulator that allows for the export of mobility traces for use in various network simulators. The developers of VanetMobiSim [14] extended CanuMobiSim by incorporating IDM as a traffic model. The authors of STRAW [15] extended SWANS by implementing a street mobility model. Saha and Johnson [16] implemented a simple mobility model directly in ns-2.

GrooveSim [17] is an integrated network and mobility simulator, but the network model has not been validated against any other well-known network simulator. The same is true for NCTUns [18]. Further, in both cases, the mobility components cannot be separated from the network components, which makes them difficult to extend further. TraNS [19] couples ns-2 with SUMO and provides for both one-way and two-way communication between them. But, as TraNS uses ns-2 for its network simulator, it inherits the scalability problems that ns-2 suffers from [20]. Plus, using two separate simulators (ns-2 and SUMO) doubles the overhead of managing the environment settings for each. Since ASH is built directly upon the scalable SWANS simulator, it does not suffer from these problems.

III. NETWORK MODEL

There are large number of network simulators currently being used in the network research community. Some of these are open source like GloMoSim, OMNet++, and ns-2, while the others are commercial like QualNet and OPNET. Of these, ns-2 is the most widely-used in the mobile ad-hoc network research community [21]. But, ns-2 suffers from problems when simulating large numbers of nodes.

SWANS [22] was developed to be a scalable alternative to ns-2 for simulating wireless networks. Based on comparisons between SWANS and other wireless network simulators, GloMoSim and ns-2, SWANS was the most scalable, the most efficient in memory usage, and consumed the least runtime. Recently, the network model in SWANS was validated against ns-2 [20]. It was shown that along with better performance, SWANS delivered similar results as ns-2, at least for the network components that were implemented in both.

IV. MOBILITY MODEL

There are four basic mobility models provided in SWANS: static, teleport, random walk, and random waypoint. These models are suitable for representing the mobility of mobile

Symbol	Definition
v_0	Desired velocity when driving on a free road
T	Desired safety headway when following other vehicles
a	Acceleration in everyday traffic
b	Comfortable braking deceleration in everyday traffic
s_0	Minimum bumper-to-bumper distance to the front vehicle
δ	Acceleration exponent

TABLE I
IDM VARIABLES [26]

ad-hoc network nodes, but they are not applicable to vehicular mobility.

Vehicular traffic models are typically classified into three categories based on traffic granularity: macroscopic, mesoscopic, and microscopic. Macroscopic models deal with traffic as flows, while mesoscopic models are concerned with the movement of whole platoons of vehicles. Microscopic models handle the movement of each vehicle in the traffic flow, thus they are the most suitable for VANET applications.

Many microscopic models have been developed. The most widespread ones are the SK model [23], the Cellular Automaton (CA) model [24], and the IDM/MOBIL model [5] [6]. Fiore *et al.* [25] evaluated the realism of several mobility models and recommended that only realistic car-following models, such as IDM, be used in VANET simulations. Thus, we have chosen to extend SWANS by implementing IDM/MOBIL to control vehicular mobility (acceleration, deceleration, and lane change). In the future, we plan to implement other car-following models in ASH to give the research community the freedom to select different models.

The Intelligent Driver Model (IDM), developed by Treiber *et al.*, is a car-following model. The general characteristic of car-following models is that the traffic state at any time is characterized by the positions, velocities, and the lane indices of all vehicles. Car-following models are also known as safety-distance models, because the inter-vehicle distance cannot be less than a specific threshold. Thus, there are no collisions when using car-following models.

IDM has several parameters that can be adapted to represent different types of drivers (Table I). For instance, aggressive drivers may have an increased desired velocity v_0 , acceleration a , and deceleration b , with a decreased safety headway T (distance between them and the car directly ahead).

Minimizing Overall Braking deceleration Induced by Lane changes (MOBIL), also developed by Treiber *et al.*, controls how vehicles change lanes. According to this model, there are two criteria that must be satisfied before any vehicle performs a lane change. The *incentive criterion* states that the targeted new lane must be more attractive than the current lane, and the *safety criterion* states that this lane change should be done safely. In other words, the lane change should be both advantageous to the driver and should not lead to an accident or discomfort to nearby drivers.

The variables used to customize MOBIL are given in Table II. The politeness factor, p , allows MOBIL to model various types of drivers, from altruistic ones who favor others over

Symbol	Definition
p	Politeness factor
b_{safe}	Maximum safe deceleration
a_{thr}	Threshold (must be below IDM parameter a)
δ_b	Bias to the right lane

TABLE II
MOBIL VARIABLES [27]

themselves to malicious drivers who only want to thwart others. Note that no matter the politeness factor, the safety criterion must always be obeyed. As the exit destination is predetermined for each vehicle at its creation time, the politeness factor for each vehicle’s driver can change dynamically as the vehicle approaches its destination exit.

V. ASH EXTENSIONS

Here we present an overview of the features included in ASH:

- Two-way communication
- Customizable highway topology
- Enhanced node model
- Implementations of IDM and MOBIL
- Implementations of Inter-Vehicle Geocast (IVG) and probabilistic IVG (p-IVG)
- Statistical and logging facilities

A. Two-Way Communication

The ability for two-way communication is the most important feature we have added in ASH. Through this feature, the application running at each vehicle can affect the vehicle’s mobility. This two-way communication is achieved by providing the application layer with two sets of primitives. The first set of primitives allows the application layer to retrieve the received messages from the network stack in the network sub-simulator. The application should provide a message handler and associate it with a specific port. When a new message arrives at the specified port, the network stack will deliver the message to the appropriate handler.

Based on the information received from the network, the application can choose to affect the vehicle’s mobility. Therefore, the application can override the IDM/MOBIL normal behavior through the second set of primitives which is a set of mobility control primitives e.g., *accelerate*, *decelerate*, *change-lane*. This means the mobility decisions will be adaptive according to the information available at the application level. In Figure 1, we show an example cooperative collision avoidance system where such control by the application is needed. The driver of car C receives an alert about a suddenly braking vehicle ahead (car A). Since car B is still outside of the safety distance of car C, IDM would not cause car C to slow down. But, the driver might choose to begin slowing down early. The two-way communication provided by ASH would allow the application running on the car C to emulate decisions that a driver might make. This feature is the most important needed feature in any simulation used by those interested in developing safety-related applications for VANET.

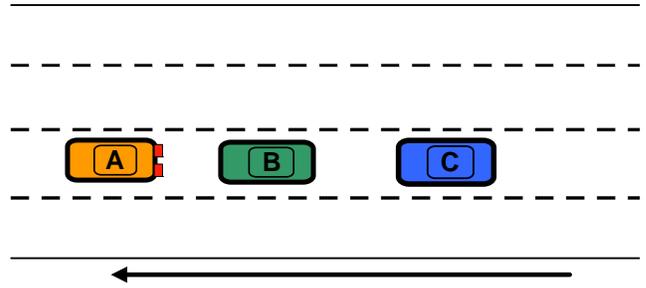


Fig. 1. Cooperative Collision Avoidance System Example

B. Highway Topology

ASH supports a simplified design of a highway segment. The segment can be configured as a one-way or two-way highway. The key point of supporting the two-way highway option is that some VANET applications require the vehicles in the opposite directions to communicate with each other. The characteristics of the highway segment can be specified in a configuration file. The supported characteristics are segment length, number of directions, number of lanes, and number of entries and exits, as well as their locations along the segment. Although this highway design is simple, it satisfies most of the requirements needed to implement scenarios in highway-based VANET applications.

In Figure 2 we show an example highway configuration file; the corresponding graphical representation is shown in Figure 3. In this example, we setup a highway segment of length ten kilometers with two directions and four lanes in each direction. The keyword `Highway` denotes the beginning of the configuration. This is followed by the specification of the highway length and number of lanes (`Length`, `Two-Ways`, and `Lanes-Count`). Once the basic characteristics of the highway have been set up, the number and position of exits and entrances can be defined. Each highway direction is specified separately, using the `Direction` and `Opposite Direction` keywords. Entries are specified using the `Entry` keyword and exits by the `Exit` keyword. Both entries and exits are defined by their ID, location (offset from the beginning of the highway), and type (left or right), defined by the keywords `ID`, `Location`, and `Type`, respectively. The highway can be set to have only one direction by setting `Two-Ways: no`. In this case, the characteristics of the `Opposite-Direction` should not be included in the configuration file.

C. Node Model

Currently in SWANS, all nodes in a simulation have the same mobility, *i.e.* nodes can all be mobile or all be static. We have enhanced the node model to allow each node to have its own mobility state, so that a mix of mobile and static nodes can be present in the same simulation. By default in SWANS, all nodes communicate. We have added the option for nodes to be silent. This results in four possible node types that can be present in the same simulation (illustrated in Figure 4):

```

Highway
{
  Length: 10          // in km
  Two-Ways : yes     // yes or no
  Lanes-Count : 4    // Number of Lanes per direction

  Direction
  {
    Exit
    {
      ID : 1
      Location : 1    // Offset in km from the highway start
      Type: R        // R=Right Exit ; L=Left Exit
    }

    Exit
    {
      ID : 2
      Location : 3    // Offset in km from the highway start
      Type: R        // R=Right Exit ; L=Left Exit
    }

    Entry
    {
      ID : 1
      Location : 1.5  // Offset in km from the highway start
      Type: R        // R=Right Entry ; L=Left Entry
    }

    Entry
    {
      ID : 2
      Location : 3.5  // Offset in km from the highway start
      Type: R        // R=Right Entry ; L=Left Entry
    }
  }

  Opposite-Direction // exists only if Two-Ways : Yes
  {
    Exit
    {
      ID : 1
      Location : 1    // Offset in km from the highway start
      Type: R        // R=Right Exit ; L=Left Exit
    }

    Exit
    {
      ID : 2
      Location : 3    // Offset in km from the highway start
      Type: R        // R=Right Exit ; L=Left Exit
    }

    Entry
    {
      ID : 1
      Location : 1.5  // Offset in km from the highway start
      Type: R        // R=Right Entry ; L=Left Entry
    }

    Entry
    {
      ID : 2
      Location : 3.5  // Offset in km from the highway start
      Type: R        // R=Right Entry ; L=Left Entry
    }
  }
}

```

Fig. 2. Example Highway Configuration File

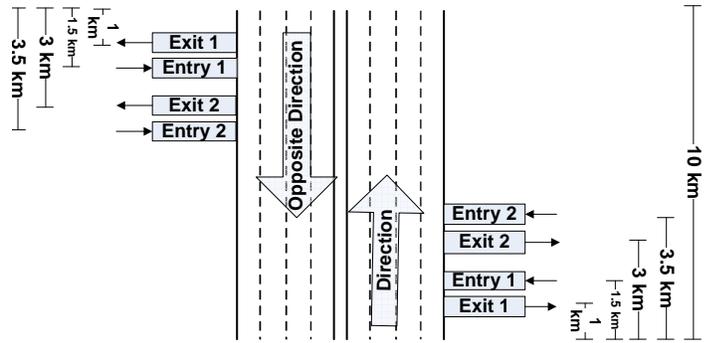


Fig. 3. Graphical Representation of Example Highway Configuration

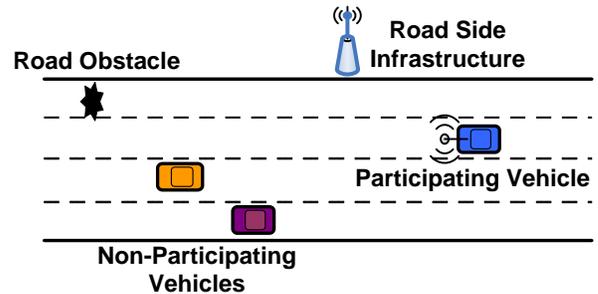


Fig. 4. Possible Nodes Types in ASH

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

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

D. Enhanced Dissemination Techniques

Because most VANET applications use flooding-based techniques to disseminate data, we implemented the Inter-Vehicle Geocast protocol (IVG) [7] in ASH. In IVG, whenever a vehicle receives a message to broadcast, it starts a timer. When the timer value expires, it broadcasts the message only if it has

not heard another node re-broadcasts the same message. The time value T_x for vehicle x is

$$T_x = T_{max} \cdot \frac{(R^\epsilon - D_{sx}^\epsilon)}{R^\epsilon},$$

where R is the transmission range and D_{sx} is the distance between vehicle x and vehicle s , the sender of the message. The authors of IVG suggest using $\epsilon = 2$ to generate a uniform timer value between $[0, T_{max}]$ where $T_{max} = 200$ ms.

In previous work [8], we developed a probabilistic version of IVG to take into account the surrounding traffic density. In probabilistic-IVG (p-IVG), when a vehicle receives a message, it first selects a random number in $[0, 1]$. Only if the selected number is less than $\frac{1}{density}$ does timer start. Otherwise, the message will not be re-broadcasted by this vehicle. As the density increases, the number of nodes that will start their timers decreases. To calculate the density, we assume that each vehicle periodically sends a beacon announcing the time, its ID, and its location. Thus, at any time, a vehicle knows the location of other vehicles within its transmission range based on received beacon messages. The density is then calculated as the number of vehicles present divided by the size of the area of interest. In p-IVG, the decision to start the timer depends upon the vehicles' density, but it can be customized to depend on any other factor according to the simulated scenario.

E. Supporting Utilities

We have also implemented statistical and logging utilities to support simulations in ASH. The utilities provide information about the simulation entities at different granularities. This information can be retrieved at the simulation level, lane level, vehicle level, or message type level. For example, to log the mobility traces for all the vehicles in the simulation, the mobility log switch should be turned on at the simulation level (`Mobility_Log = true`). The mobility traces of all vehicles will be stored in a single file, set by `Mobility_Log_File = filename`. To log the mobility traces for a specific vehicle, the mobility log switch for this vehicle should be turned on when the vehicle is created (`Vehicle.Mobility_Log = true`). In this case the log file in which the vehicle's mobility traces will be stored should have the vehicle ID as part of its name (`Vehicle.Mobility_Log_File = filename + Vehicle.ID`). Similarly, any message type can be logged at either the vehicle level or the simulation level. Also, the data can be projected to the time domain, meaning that the information can be provided for the whole simulation time, any simulation period, or a specific simulation instance. The provided utilities are customizable so they can be turned on or off according to the user's needs.

The statistics utility provides statistics about all possible events that can occur in the simulation. For example, it keeps track of the number of messages sent or received for each message type. This count is maintained at both the vehicle level and at the simulation level. Moreover, the statistics utility keeps track of the number of vehicles in the simulation and how many of them are in each lane. This utility

works automatically without the needing to be turned on or off. All of the statistics are stored in variables and arrays that are accessible system-wide. For example, the number of current vehicles in direction `dir` and lane `lane` is stored in `Vehicles_Per_Lanes[dir][lane]`. As another example, the variable `Vehicles.[V_ID].Messages_Sent` stores the total messages sent by vehicle `V_ID`.

VI. CONCLUSION

We have presented ASH, comprising our enhancements to the SWANS network simulator to provide feedback between the application layer and mobility model, allowing for two-way communication. We have also implemented the IDM/MOBIL mobility model as well as enhancing SWANS' node model and adding a highway topology. In the future, we plan to implement additional mobility models and add a visualization component.

The source code of ASH will be made available through our project website: <http://www.cs.odu.edu/vanet/>

VII. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant CNS-0721586.

REFERENCES

- [1] J. Ott and D. Kutscher, "Drive-thru Internet: IEEE 802.11b for "automobile" users," in *Proceedings of IEEE INFOCOM*, 2004.
- [2] M. Abuelela and S. Olariu, "Zipper: a zero-infrastructure peer-to-peer system for VANET," in *Proceedings of the ACM Workshop on Wireless Multimedia Networking and Performance Modeling (WMuNeP)*, Crete Island, Greece, 2007, pp. 2–8.
- [3] K. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla, "First experience with CarTorrent in a real vehicular ad hoc network testbed," in *Proceedings of IEEE MOVE*, May 2007.
- [4] <http://jst.ece.cornell.edu/>.
- [5] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, p. 1805, 2000.
- [6] A. Kesting, M. Treiber, and D. Helbing, "MOBIL: General lane changing model for car-following models," in *Proceedings of the Transportation Research Board Annual Meeting*, Jan. 2007.
- [7] A. Bachir and A. Benslimane, "A multicast protocol in ad hoc networks inter-vehicle geocast," in *Proceedings of IEEE VTC-Spring*, 22-25 April 2003, pp. 2456–2460.
- [8] K. Ibrahim and M. C. Weigle, "CASCADE: Cluster-based accurate syntactic compression of aggregated data in VANETs," *In Submission*, 2008.
- [9] F. Karnadi, Z. H. Mo, and K.-C. Lan, "Rapid generation of realistic mobility models for VANET," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2007, pp. 2506–2511.
- [10] <http://sumo.sourceforge.net>.
- [11] <http://isi.edu/nsnam/ns>.
- [12] <http://www.scalable-networks.com>.
- [13] <http://canu.informatik.uni-stuttgart.de>.
- [14] J. Härrä, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: Generating realistic mobility patterns for VANETs," in *Proceedings of ACM VANET*, Los Angeles, CA, 2006, pp. 96–97.
- [15] D. R. Choffnes and F. E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks," in *Proceedings of ACM VANET*, Cologne, Germany, 2005, pp. 69–78.
- [16] A. K. Saha and D. B. Johnson, "Modeling mobility for vehicular ad-hoc networks," in *Proceedings of ACM VANET*, Philadelphia, PA, 2004, pp. 91–92.

- [17] R. Mangharam, D. S. Weller, D. D. Stancil, R. Rajkumar, and J. S. Parikh, "GrooveSim: a topography-accurate simulator for geographic routing in vehicular networks," in *Proceedings of ACM VANET*, Cologne, Germany, 2005, pp. 59–68.
- [18] S. Y. Wang, C. L. Chou, Y. H. Chiu, Y. S. Tzeng, M. S. Hsu, Y. W. Cheng, W. L. Liu, and T. W. Ho, "NCTUns 4.0: An integrated simulation platform for vehicular traffic, communication, and network researches," in *Proceedings of IEEE VTC-Fall*, 2007, pp. 2081–2085.
- [19] M. Piorkowski, M. Raya, A. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "TraNS: Realistic Joint Traffic and Network Simulator for VANETs," *ACM MC2R*, 2007.
- [20] F. Kargl and E. Schoch, "Simulation of MANETs: A qualitative comparison between JiST/SWANS and ns-2," in *Proceedings of the International Workshop on System Evaluation for Mobile Platforms (MobiEval)*, San Juan, Puerto Rico, 2007, pp. 41–46.
- [21] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET simulation studies: The incredibles," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 4, pp. 50–61, 2005.
- [22] R. Barr, Z. Haas, and R. van Renesse, *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks*. CRC Press, 2005, ch. Scalable Wireless Ad Hoc Network Simulation, pp. 297–311.
- [23] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," Ph.D. dissertation, Mathematisches Institut, Universität zu Köln, 1998.
- [24] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *J. Phys. I France*, vol. 2, p. 2221, 1992.
- [25] M. Fiore, J. Haerri, F. Filali, and C. Bonnet, "Understanding vehicular mobility in network simulation," in *Proceedings of the 1st IEEE international Workshop on Mobile Vehicular Networks (MoVeNet)*, in conjunction with *IEEE MASS 2007*, Pisa, Italy, Oct. 2007.
- [26] M. Treiber, "Longitudinal traffic model: IDM," <http://vwisb7.vkw.tu-dresden.de/~treiber/MicroApplet/IDM.html>, Apr. 2006.
- [27] —, "Lane change model: MOBIL," <http://vwisb7.vkw.tu-dresden.de/~treiber/MicroApplet/MOBIL.html>, Feb. 2007.