Towards more Realistic Simulations of Ad-hoc Networks - Challenges and Opportunities

Steffen Moser and Frank Slomka
Institute of Embedded Systems/Real-Time Systems
Faculty of Engineering and Computer Science
Ulm University
89069 Ulm, Germany
Email: {steffen.moser | frank.slomka}@uni-ulm.de

***Abstract*—Vehicular Ad-hoc Networks are a very interesting self-organizing communication system that can make the traffic on our roads much safer. A lot of aspects of the behavior of these systems are studied using network simulation frameworks. While the upper communication layers of Vehicular Ad-hoc Networks are mainly simulated according to their specifications, in this paper we will highlight weak points in these simulations. They occur due the fact that the layers below the link layer are considered very rarely and are also almost neglected by the commonly used network simulation frameworks. This can potentially lead to inaccurate simulation results which can be fatal when studying safety relevant applications or application with real-time constraints. In this work in progress paper we are going to propose an approach to get this under control.**

I. INTRODUCTION

Giving vehicles the ability to exchange information with each other can be a chance to reduce the number of car accidents on our streets.

Exemplary applications would be to allow cars to send warning messages about hazardous street conditions like sudden ice or oil, but also an enhancement of a driver's range of sight especially for badly visible intersections. Also non-safety related applications are possible, for example achieving a more efficient flow of traffic given by the ability to exchange information about traffic load and traffic jams with other cars.

Wireless ad-hoc networks are a very actively researched topic as they offer a possibility to allow vehicles to communicate with each other without the need of setting up an expensive wired infrastructure like base stations that are regularly used for mobile communication. When applying the idea of wireless ad-hoc networks to vehicle communication, they are often called vehicular ad-hoc networks (VANETs). VANETs can be characterized as self-organizing distributed systems of vehicles communicating as well as with each other (Car-to-Car or C2C) but also with road-side units (Car-to-Infrastructure or C2I) like sensors mounted at the street poles or receivers at traffic lights.

The automotive industry seems to be seriously interested in deploying VANET applications in future automobiles. The foundation of the Car-2-Car Communication Consortium (C2C-CC) and the high number of ongoing research activities with participation of automotive manufacturers point this up. IEEE has released the draft standard 802.11p exclusively for C2C and C2I. 802.11p is closely based on Wireless LAN according 802.11a, though it works on its own frequency band in the region of 5.9 GHz.

A very typical attribute of VANETs in contrary to other ad-hoc network applications is their highly dynamic behavior. This makes VANETs very challenging and they are therefore one of the most intensively studied applications of ad-hoc networks. The most parts of VANET research are done via network simulations. This is mainly due to two well-known reasons: Large scale experiments would require a high number of vehicles equipped with a prototype of a communication device which is not only more expensive than simulations, but also less flexible at early development stages. Another advantage of simulations is that they allow studying the direct impact of distinct parameters as simulations deliver reproducible results. While keeping, for example, environmental parameters like the behavior of the radio channel or the distribution of the vehicles on the street constant, the influence of other parameters, like the algorithm used to route packets through the network, can be studied. Especially in such a complex distributed system this would be not possible when working only with experiments.

As the advantages of using simulations in VANET research are undeniable, considering their accuracy is absolutely necessary.

The simulation frameworks commonly used to study the behavior of VANETs are derived from the field of wired packet switched networks. This means that the typical upper layers of a network, like queuing and routing packets, doing error correction and retransmission, acknowledging packets and establishing connections and so on, are regularly simulated according to the specifications of a given network system. However, when it comes to highly dynamic wireless networks like VANETs, additional aspects are getting more important: For example, the mobility of the nodes, the processing of the wireless signals and the behavior of the time-variant radio channel. Therefore the mentioned simulation frameworks have been extended by code to allow mobile nodes moving around according to mobility models and to simulate also wireless networks, but when looking at the implementation of the wireless communication part in commonly used network simulation frameworks more closely, one can notice that the behavior of the system below the media access control layer (MAC) is simulated in quite a simplified way only:

- The simulation of the physical layer, i.e. the whole signal processing, is mostly omitted which means that only estimations are done regarding the possibility to receive a data frame and regarding the bit error rate that can be expected. These estimations are based on heuristics for a specific networking protocol like the above mentioned IEEE 802.11p, which makes it difficult to do research on alternative MAC mechanisms using the available simulation frameworks, because exactly temporal characteristics of the signal are not available. Alternative MAC mechanisms can be interesting for applications that need real-time communication.
- Another challenge is the accuracy of the radio channel simulation. The radio channel is regularly simulated by doing estimations about free-space path loss, which is only a function of the sender's and receiver's position at the beginning of the transmission process. The influence of the environment is completely neglected. As this might be an acceptable approximation for highway scenarios in a flat free-space environment, it has been shown that the influence of buildings on the radio channel is quite noticeable and should be especially considered when simulating inner-city scenarios.

Both aspects show that there is a need for a further development of network simulation frameworks in order to allow to run more precise simulations and to explore new technologies in the area of car-to-car and car-to-infrastructure communication for time-critical applications.

In this paper we present an approach to include realistic physical layer simulation into a simulation framework used for VANET research. This also allows achieving more realistic channel simulations. Therefore, in the next chapter we will go into the depth of typically used simulation frameworks and show how the physical layer and the radio channel are simulated. The third chapter shows the requirements we have on a simulation framework. In the fourth chapter we come up with our proposal and discuss related work in the field of VANET simulation frameworks. We conclude with a summary and open questions in the fifth and last chapter.

II. SIMULATIONS OF VANETS

This section gives an overview about typical properties of VANETs simulations. Therefore at first common basics about network simulation frameworks are presented. After then two common simulation frameworks are described in a more detailed way and the differences between them are elaborated. This helps to understand the reasons for the problems which are then presented in the following sections.

A. *Basics of Network Simulation Frameworks*

In this section, a short overview about the basics and the main functionality of typical network simulation frameworks used for VANET research are given.

The most important attribute that all simulation frameworks that are used for simulating VANETs have in common is that they are implemented based on the idea of a discrete event simulator. In short they work as follows: The system state is stored in variables that can change whenever an event is triggered. When, for example, a network node sends a message, the sending event is put into an event queue. The queued events are scheduled at discrete points in time. The processing of an event itself happens instantly which means that no simulation time elapses during its execution. If necessary, the simulation time is advanced explicitly by using specific commands, for example to simulate the time that was needed for sending the message. A more detailed discussion of discrete event simulations is given by Peschlow and Martini in [1].

The typical simulation framework used for VANET simulations consists of the following main parts:

- The discrete event simulator, i.e. the simulator kernel. It is responsible for queuing and scheduling events and managing local and global simulation times.
- The mobility model and its simulation. Its task is to move the network nodes according to simulate a realistic flow of traffic, e.g. accelerating and braking cars, intersections, traffic lights and traffic jams.
- The communications model and its simulation. It is responsible for the radio communication. This is the network stack with the common layers beginning with the application layer down to the physical layer. Also the simulation of the communication channel belongs to this part.
- Components to manage the simulations, i.e. reading configurations, distributing large simulations on cluster computers, collecting statistics and doing analysis.

There are also differences between various simulation frameworks. One noticeable attribute is the question how the simulator is actually accessible by the user. One sort of simulators is working language based which means that the user can implement a network application in a common programming language (e.g. C++ or Java) and can access typical networking methods supplied by a library which are then executed within a simulation kernel. The other sort of simulators works by defining an own simulation language which directly allows to setup network scenarios.

B. *Commonly used Simulation Frameworks*

In this section we introduce two of the commonly used simulation frameworks for VANETs.

One of the most widely used frameworks for network simulation in VANET research is The Network Simulator ns-2 [2]. As the first version appeared 1989, it has quite a long history. It is still actively developed and offers a large variety

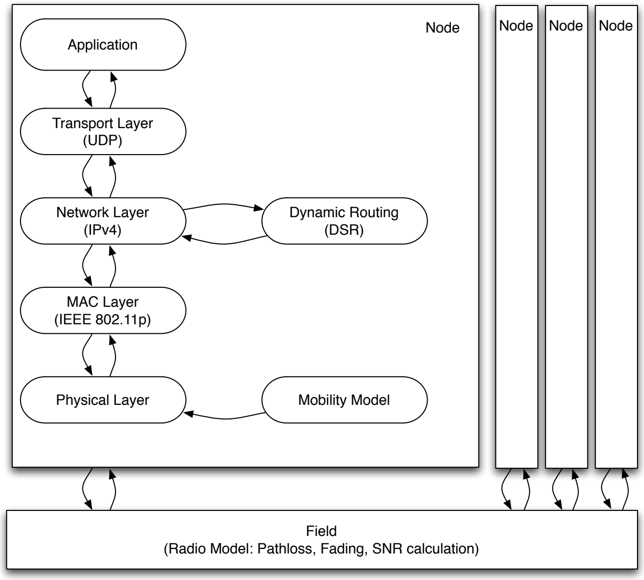

Fig. 1. Network layers available in JiST/SWANS [4]

of network protocols, routing algorithms and media access mechanisms. It was, like other network simulators, originally written for the simulation of wired networks, but has been extended for wireless networks in the meantime as described in [3]. ns-2 uses a combination of both access methodologies explained in the previous section. The user can implement network applications directly in C++ and use then simulation libraries instead of normally used library calls. In ns-2 it is also possible to describe simulation scenarios in the OTcl language. This fact, together with the large number of available features and additionally available extensions, make the simulator quite suitable for a wide variety of uses, but it is outperformed by newer lightweight simulation frameworks.

A much newer approach is JiST/SWANS which is introduced in [4]. As the name leads to the assumption, it consists of two parts: JiST is the abbreviation for “Java in Simulation Time” and is the actual discrete event simulation engine. This is completely written in Java. To enable the usage of simulation time, JiST comes with a Java bytecode rewriter to allow the discrete event simulation. It can be seen as a transparent layer between the simulated application and the Java Virtual Machine. SWANS is the “Scalable Wireless Ad Hoc Network Simulator” which is based on JiST and was written especially for the simulation of ad-hoc networks, so it lacks a lot of other protocols and mechanisms that are available in other frameworks. JiST/SWANS simulates the typical layers available in an IP based communication system as shown in figure 1.

The layers can be accessed also from self-written Java based applications. In [5] it was demonstrated that JiST/SWANS is very efficient regarding computation time and memory usage compared to other simulation frameworks like ns-2. Even for very large simulations JiST/SWANS scales almost linearly as shown in [6], where up to 13,100 nodes have been simulated on an area of 10 x 16 km. This is mainly due to its optimization for the special purpose of simulating ad-hoc network scenarios. In concrete, JiST/SWANS strictly avoids marshaling and demarshaling of any data sent through the simulated network layers. Packets are handled as objects, and only references to these objects are passed through all of the communication layers. Further optimizations are done in the area of the field simulation: JiST/SWANS estimates which vehicles are actually near enough to the sender so that they have a chance to receive the signal. Only these nodes are then considered in the calculation of the actual signal and noise power which can be expected.

C. Simulation of the Physical Layer

The physical layer of a communication device is the interface between the analog electrical signals and the digital data. So it is responsible for processing a bit stream of data coming from the upper layer to form an analog signal which can be sent via one or more antennas and vice versa for the received signal.

Dependent on the physical layer specification of a network protocol, a lot of mathematical operations of the field of telecommunications engineering are necessary for this, for example, modulation and demodulation, FFT and inverse FFT, auto- and cross-correlation.

In order to reduce the computational costs of network simulations and to avoid modeling the whole signal forming and decoding process the introduced simulation frameworks do not perform these tasks. Instead of this, the simulation frameworks abstract from the physical signal. The idea is to pass the digital data from the sender directly to these nodes that would be able to gather the information in reality. This is estimated by calculating the signal power that can be expected at the receivers’ positions for a given sender’s position according to the channel model which is discussed in detail in the next section. After then, the simulator compares the expected signal to noise ratios with the data-sheets of the used hardware, which allows the simulator to decide whether a signal would be able to be detected at a specific receiver or not. Additionally, the probability of bit errors is calculated, also based on the signal to noise ratio and a lookup table which contains experienced data about the bit error rate that can be expected for a given signal to noise ratio. The actual process of sending and receiving data at the physical layer is modeled as follows: Each node has a status flag which represents the mode the node is currently in, so it can be simulated if a node is idle, currently sending or receiving. The setting of the status flag is changed according to the amount of time a specific process needs as per specification.

JiST/SWANS goes even one step ahead than other simulators. As already stated, JiST/SWANS strictly tries to avoid serializations and deserializations of packet data due to performance reasons. This does not only happen when a packet is passed from layer to layer within the communication stack of a simulated network node. In fact, this paradigm also holds for communication between different nodes: The receivers of a signal that has been sent out do not get a copy of the bit stream, but only a pointer to the object that has been created at the sender. This is one instrument to further reduce the computation overhead, but it also shows how much the simulation frameworks abstract from the real signal processing.

D. Simulation of the Radio Channel

In contrary to wired computer networks, communication in wireless systems is much more unrealiable. This section will give an overview about the modelling of the radio channel in the simulation frameworks.

When studying, for example, the behavior of packet queues in routers or switches in local wired networks, then the influence of the cable on the electrical signal can be ususally neglected. Especially if the network topology is such that several nodes do not share a cable, one can assume that all bits sent from a node are received at the port of the switch - at least as long as the system is free of failures and is run within its specifications.

This is totally different in wireless networks. Here, the medium used for communication is not only shared by the nodes of the network, but also less reliable in general due to its nature. This means that there are various physical effects that occur when radio waves propagate from the sender to the receiving nodes, especially when the waves interact with matter. Waves spread into space, get reflected specularly at walls, are partly absorbed by all kinds of matter, are scattered at rough surfaces or diffracted at edges. This leads to the phenomen of multi-path propagation, i.e. the signal power arrives at the receiver on different paths with different lengths and therefore spreads in time.

Multi-path propagation is one cause for interference effects which can lead to a very strong fluctuating signal power even to the point of a total signal loss in the worst case.

As stated in the previous section about the physical layer, the available simulators estimate the expected signal to noise ratio that can be measured at the receiver. This is done by using one of the implemented channel models. As described in [7], a channel model can be broken down to the following parts:

- The model of the sending antenna,
- the path-loss model,
- the fading model,
- and the model of the receiving antenna.

The path-loss model describes the long-term loss of signal power when sender and receiver move apart. The simplest path-loss model is the free-space path-loss model. It considers only the loss of power per area which occurs due to spreading of the electromagnetic waves into free space. This results in the following simple function that only depends on the Euclidian distance r between sender and receiver:

$$P_{receiver}(r) = P_{sender} \cdot \frac{1}{4\pi r^2} \tag{1}$$

Usually, the gain of the sender's and receiver's antenna ($G_{sender}\ G_{receiver}$) as well es wavelength λ dependent antenna effects are also considered. This results in

$$P_{receiver}(r, \lambda) = P_{sender} \cdot G_{sender} \cdot G_{receiver} \cdot (\frac{\lambda}{4\pi r})^2 \tag{2}$$

This is also called Friis' Transmission Formula [8]. This is the default path-loss model used in many network simulations. There are a bit more complex models available in simulators, too. For example, the two-ray ground model considers also interference effects caused by the part of the wave which is reflected by the road surface, but not by actually calculating these effects. They are rather approximated based on simple trigonometric assumptions as discussed in [9]. Like the free-space path loss model, the two-ray ground model depends only on the wavelength and the distance between sender and receiver. The real surface of the road is not considered.

In order to make the channel model complete, a fading model is added to the path-loss model. Fading occurs due to the movement of the nodes and the consequently changing propagation conditions (i.e. when getting in or leaving the line of sight between sender and receiver, reflections, absorptions, and so on), but it is usually calculated only as a statistical distribution without any input of environment. Rayleigh and Rician fading are usually used.

Using this channel model, the simulator calculates the signal power which can be expected at a specific receiving node. The signal power of other senders that are simultaneously active are summed up and result in the noise (additive noise model) which the receiver has to struggle with.

Exactly like in the area of physical layer simulation, the simplifications that are implemented in network simulators go far. It was shown in several publications, for example in [10], that these simplified path-loss calculations over-estimate the signal quality a lot, especially in inner-city scenarios where the signal is strongly influenced by buildings. At least for simulations in these areas radio channel models are needed that consider the environment and its influence on radio wave propagation. One possible approach are models based on ray tracing, a technique which has its origin in the field of computer graphics. It is used to generate photo-realistic images out of descriptions of a scene by simulating the propagation of light. When modifying the physical effects according to the wavelengths used for radio communication, ray tracing can be used to simulate the propagation of radio waves between sender's and receiver's antenna with considering the interactions with matter.

In [11] we showed in a field test in how far the free-space path-loss overestimates the measured signal power in an typical inner-city scenario with lots of regular buildings. It also shows that a ray tracing based approach predicts the measured signal power in a much more precise way. Figure 2 depicts the measured signal intensities at various line-of-sight and non-line-of-sight positions and compares them to the values calculated according to a free-space pathloss model and a ray tracing based approach.

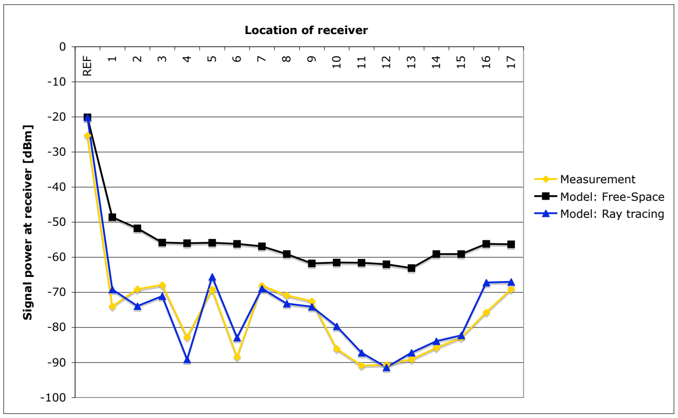

Fig. 2. State-of-the-art methods of VANET simulators overestimate the signal level in urban areas as they completely ignore the environmental effects on the radio propagation channel. We have shown that considering obstacles like buildings using a ray-tracing based approach to calculate the average receiving power at various positions in a scenario increases precision dramatically.

III. Motivation

As we have shown, available simulation frameworks for VANETs have limitations caused by their abstraction level from the physics of such a complex system like VANETs. Nevertheless, they are used and a lot of research is done, for example in the areas of finding optimal routing algorithms and procedures to aggregate sensor data to reduce the amount of information that has to be sent via the VANET, but also for research in security and privacy issues.

In this section we explain why the capabilities of the available network simulators are not sufficient from our point of view.

- On the one hand, there are the poor radio channel models. These fail especially in complex inner-city scenarios. One solution would be to simulate just the radio channel more precisely, like described in [11]. The problem hereby is that only very little parts of the results of the more precise radio channel simulation could actually be used by the physical layer: It can only handle scalar values for signal and noise power due its implementation in nowadays' simulation frameworks. The remaining information of the channel's influence on the signal would get just lost.
- On the other hand, some of the VANET applications that are being discussed are time-critical. For example, a driver assist system which has the task to warn the driver about events arriving in invisible areas of the road like the end tail of a traffic jam in a bent, has to receive the warning message early enough that the driver is able to brake in a safe way.

  Both, academic research groups as well as the automotive industry focus on IEEE 802.11p. This is the link layer protocol which has been proposed for communication in VANETs. Like in all other protocols based on IEEE 802.11, the MAC mechanism of 802.11p is designed as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). As a competitive media access scheme, CSMA/CA cannot give any real-time guarantees. The time which elapses until a node is allowed to send is fully dependent on the current channel usage. As Bilstrup et al. showed in [12], this can be problematic for some of the discussed applications. They propose Self-Organizing Time Division Multiple Access (S-TDMA) as an alternative approach for VANETs. This protocol has a short competition phase only at the beginning when new nodes choose their time-slot. After then, the communication happening in fixed slots.

  We would like to evaluate also other approaches on the MAC layer. Besides variants of Self-Organizing TDMA, also Code Division Multiple Access (CDMA) can be promising regarding real-time communication. But for studying more complex protocols, there is the need of simulations that calculate the actual physical signal over time.

Both aspects, the more precise channel simulation as well as the simulation of accurate timing aspects would benefit a lot from a physical layer simulation that effectively simulates the processing of the signal.

IV. Coupling Discrete Event Simulators with Physical Model

This section introduces our approach to improve the above described problem. It is currently being implemented and hence work in progress.

The basic idea is to use one of the network simulation frameworks but to swap out both, the physical layer and the channel model, to a Matlab/Simulink based implementation. This is due to the fact that the Matlab Communications Blockset and the Simulink Communications Toolbox offers all of the functions that are needed to do the signal processing. So they do not have to be reimplemented.

We chose JiST/SWANS as the network simulation framework, although our approach with the signal processing simulation will compensate most of the performance improvements which were originally delivered by JiST/SWANS. The reason for chosing it is the simple fact that the code is quite well documented.

Physical layer simulations especially for VANETs using the Simulink Communications Toolbox have been described in literature before, for example in [13]. Their approach lacks the connection to the network simulation environment to do simulations at large.

The challenges that emerge are the follows:

1) The discrete event simulator must be combined with the coninuous real world of the signal.
2) The discrete event simulator has to be coupled to the physical layer and radio propagation simulation in an efficient way.

We address the first challenge as follows: Assuming one specific network node N queues an event to send out data. As soon as this event is executed, the Simulink environment calculates the signal which would leave the node's antenna. A sampled form of the time-continuous signal is stored temporarily within the Simulink framework. In a very naive approach, there would have to start decoding attempts for all other nodes in the field. This would produce enormous costs.

Instead of this, the network simulator informs the Simulink environment about a) the nodes which are currently receiving data (addressed for them) and b) the nodes that are positioned near enough to the sending node N so that they can just barely gather the signal in the best propagation conditions. Only for the nodes fulfilling both conditions a) and b), a decoding function is started in Simulink due to node N's transmission. This is done by calculating the influence of the radio channel for all signal paths between N and each receiving node. For each of the receiving nodes, the arriving signal from node N is added to all other signals that this node can receive at the same time. After then, for each of the receiving nodes the actual decoding process is initiated. The resulting digital information is then passed back to JiST/SWANS.

This approach is general enough to allow also communication protocols where one receiver can get information from more than one sender at the same time and frequency, like in CDMA based systems.

The second challenge is about the co-operation of physical layer and radio channel simulation in Simulink. As the actual sending of data does not happen in zero-time, it would need an enormous amount of computation time when simulating the channel behavior for every sampled value of the transmitted signal. The usual procedure as it is also described in the literature is to simulate sending an infinitely short pulse and to record the response that the channel produces. The response is stored, and the actual sent data signal is then convoluted with the pulse response. For a higher accuracy it is easily possible to repeat the pulse response simulation during the sending process in an arbitrary frequency. This allows to consider also very fast moving nodes like vehicles on a highway scenario.

Without any doubt, these more precise simulations will cause a lot of computation overhead compared to network simulations as they are normally done in VANET research. But we have to consider that the simulation of accurate radio propagation by using ray tracing methods has made extraordinary progress by exploiting the possibilities offered by modern graphics processors. This has been shown recently in [14].

V. CONCLUSION

In this paper we have shown that even recent network simulation frameworks neglect effects that are occuring in wireless networks almost totally. We think that the simulation frameworks have to be improved with two goals: The first one is that the physical layer should be modeled in a more general way to be more flexible for newer approaches. This includes to do signal processing. The second goal should be to support more realistic channel simulations which is especially important when simulating traffic in inner-cities. The second goal requires the first one to be realized. Of course, both goals are in contrary to the ambition of reducing the computation time that is needed for running a simulation. But as it has been explained that the extensions can be implemented as an adjustable trade-off between accuracy and run-time. Nowadays, it is possible to do accurate simulations of radio wave propagation in an acceptable amount of time by using graphics processing units.

This paper is still work in progress, as our implementation work has not been finished, yet. Therefore we cannot give any results at the current point in time.

REFERENCES

[1] P. Peschlow and P. Martini, "A discrete-event simulation tool for the analysis of simultaneous events," in *ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 1–10. [Online]. Available: http://portal.acm.org/citation.cfm?id=1345281

[2] "The Network Simulator ns-2," Dec. 2009. [Online]. Available: http://www.isi.edu/nsnam/ns/

[3] D. B. Johnson, "Validation of Wireless and Mobile Network Models and Simulation," in *Proceedings of the DARPA/NIST Workshop on Validation of Large-Scale Network Models and Simulation*, Fairfax, VA, May 1999. [Online]. Available: http://www.monarch.cs.rice.edu/monarch-papers/darpa99.ps

[4] R. Barr, "An efficient, unifying approach to simulation using virtual machines," Ph.D. dissertation, Cornell University, 2004. [Online]. Available: http://jist.ece.cornell.edu/docs/040517-thesis.pdf

[5] E. Schoch, M. Feiri, F. Kargl, and M. Weber, "Simulation of Ad Hoc Networks: ns-2 compared to JiST/SWANS," in *First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SimuTools)*, 2008.

[6] M. Feiri, "Simulation von VANETs mit JiST/SWANS," Master's thesis, Ulm University, 2007.

[7] S. R. Saunders and A. Aragn-Zavala, *Antennas and Propagation for Wireless Communication Systems*, 2nd ed. Wiley, 2007.

[8] H. Friis, "A note on a simple transmission formula," in *Proceedings of IRE, vol. 41*, 1946.

[9] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[10] I. Stepanov and K. Rothermel, *Simulating Mobile Ad-Hoc Networks in City Scenarios*, 2006, ch. Session 1: Mobile Ad-Hoc Networks, pp. 1–12.

[11] S. Moser, F. Kargl, and A. Keller, "Interactive Realistic Simulation of Wireless Networks," in *Proceedings of the IEEE/EG Symposium on Interactive Ray Tracing 2007*, 2007.

[12] K. Bilstrup, E. Uhlemann, E. G. Str?m, , and U. Bilstrup, "On the Ability of the 802.11p MAC Method and STDMA to Support Real-Time Vehicle-to-Vehicle Communication," in *EURASIP Journal on Wireless Communications and Networking vol. 2009, Article ID 902414*, 2009. [Online]. Available: http://downloads.hindawi.com/journals/wcn/2009/902414.pdf

[13] S. Shooshtary, "Development of a MATLAB Simulation Environment for Vehicle-to-Vehicle and Infrastructure Communication Based on IEEE 802.11p," Master's thesis, University of Gävle, Department of Technology and Built Environment, 2008.

[14] A. Schmitz, T. Rick, T. Karolski, T. Kuhlen, and L. Kobbelt, "Simulation of Radio Wave Propagation by Beam Tracing," in *Eurographics Symposium on Parallel Graphics and Visualization*, J. Comba, K. Debattista, and D. Weiskopf, Eds., 2009.