# Enabling Accurate Cross-Layer PHY/MAC/NET Simulation Studies of Vehicular Communication Networks

Jens Mittag, Stylianos Papanastasiou, Hannes Hartenstein, *Member, IEEE*, Erik G. Ström, *Senior Member, IEEE*

***Abstract*—Vehicle-to-vehicle and vehicle-to-roadside communications is required for numerous applications that aim at improving traffic safety and efficiency. In this setting, however, gauging system performance through field trials can be very expensive especially when the number of studied vehicles is high. Therefore, many existing studies have been conducted using either network or physical layer simulators; both approaches are problematic. Network simulators typically abstract physical layer details (coding, modulation, radio channels, receiver algorithms, etc.) while physical layer ones do not consider overall network characteristics (topology, network traffic types and so on). In particular, network simulators view a transmitted frame as an indivisible unit, which leads to several limitations. First, the impact of the vehicular radio channel is typically not reflected in its appropriate context. Further, interference due to frame collisions is not modeled accurately (if at all) and, finally, the benefits of advanced signal processing techniques, such as interference cancelation, are difficult to assess. To overcome these shortcomings we have integrated a detailed physical layer simulator into the popular NS-3 network simulator. This approach aims to bridge the gap between the physical and the network layer perspectives, allow for more accurate channel and physical layer models, and enable studies on cross-layer optimization. In this paper, we exemplify our approach by integrating an IEEE 802.11a and p physical layer simulator with NS-3. Further, we validate the augmented NS-3 simulator against an actual IEEE 802.11 wireless testbed and illustrate the additional value of this integration.**

***Index Terms*—Vehicle-to-vehicle communication, Vehicle-to-infrastructure communication, 802.11p, Physical layer simulation, Network layer simulation, Cross-layer.**

## I. Introduction

The potential of communicating vehicles which exchange messages to enhance safety while on the road has been considered for several decades and placed in sharp research focus since the early 1990's. The success and significant evolution of the IEEE 802.11 standard family has prompted standardisation bodies both in Europe and the U.S. to adopt it, through the p amendment, as a basis for a vehicular communications framework. Significantly, such a framework has now been completed to a sufficient degree and is scheduled for use in current and future field operational tests. Such tests, however, can be very expensive especially when deploying a significant number of participating vehicles, a fact which affects both the scope and frequency of experimentation and research. Furthermore, in such field operational tests it can be very difficult – and expensive – to control environmental impact and perform systematic assessments. Intuitively then, most vehicular network studies in the past have been based on simulations and one may also expect future field operational tests to be performed only after having shown promising results in simulations.

Typically, existing simulation studies in the area of vehicular communications have used either network or physical layer simulators in order to optimize the performance of the physical (PHY), medium access control (MAC) and network (NET) layers. While most of the popular network simulators, e.g., NS-2, NS-3, QualNet and OMNET++, have been developed to enable performance studies on communication networks, traditional physical layer oriented simulators, such as MATLAB [1] and the IT++ framework [2], allow the study of signal processing algorithms under specific channel conditions and usually evaluate the performance achievable by a single communications link. As a consequence, network simulators abstract the physical layer and consider the frame/packet as an indivisible unit, thereby ignoring (channel) effects on individual bits as well as specific coding and signal processing details. Similarly, physical layer simulators do not readily consider network characteristics nor do they reflect the functionality of medium access or network layer protocols.

The implied separation of these two aspects leads to several drawbacks. First, the impact of a double-selective (time and frequency varying) communications channel, such as the one at the 5.9 GHz frequency band as planned for vehicular communications, is not reflected in its appropriate context by existing network simulators, but only, possibly, through a statistical dimension. Second, interference, either due to frame collisions on the same communications channel or due to transmissions on adjacent channels, is not modelled accurately, if at all, in network simulators. Finally, it is currently not possible to investigate in a networking context the benefit of advanced signal processing techniques and their implications for the medium access or network layer – popular network simulators do not consider, for such a purpose, a detailed enough representation of a frame's transmission.

With this paper, we aim to illustrate and motivate the aforementioned drawback of having two different and separated perspectives by taking a detailed look at the protocol

J. Mittag is with the Institute of Telematics, Karlsruhe Institute of Technology, Germany (email: jens.mittag@kit.edu)

S. Papanastasiou is with the Dept. of Signals and Systems, Chalmers University of Technology, Sweden (email: stypap@chalmers.se)

H. Hartenstein is with the Steinbuch Centre of Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany (e-mail: hartenstein@kit.edu)

E. G. Ström is with the Dept. of Signals and Systems, Chalmers University of Technology, Sweden (email: erik.strom@chalmers.se)

stack through the eyes of both worlds. By discussing the limitations due to this separation we further aim to increase the awareness within the research community and want to point out the significant benefits that can be achieved by merging the perspectives.

Recently, several research efforts have attempted to address these issues, at least partially, as well. Judd et al. [3] have developed a wireless network emulator which simulates the fine-grained effects of a mobile wireless communication channel by multiplexing the antenna in-/outputs of commercial communication systems with a software-controlled digital signal processor. Thereby, controlled and repeatable wireless experiments of e.g., up to 16 IEEE 802.11-based communication systems operating at 2.4 GHz are supported. Further, with respect to the evaluation of advanced signal processing techniques, the authors of [4]–[6] have proposes several software defined radio platforms that provide the ability to implement (and emulate) the physical layer entirely in software by using dedicated hardware only for the radio frontend.

The above works propose emulation of either only the wireless communication channel or the physical layer but not both concurrently, and, further, are either expensive or difficult to use for studies of vehicular communication networks. This paper builds on and extends the preliminary results of [7] and proposes to perform detailed simulation of both the physical layer and the wireless channel entirely in software, and integrate that aspect into an existing general-purpose network simulator, namely NS-3. The merge of the network and physical layer simulation aspects allows more accurate simulations of vehicular communication networks. Increased accuracy is crucial, for instance, in studies of microscopic active safety scenarios, where it is required to know precisely when and where a message cannot be received successfully. Without such exact understanding of why messages are not received, it is difficult to design robust and reliable vehicular communication systems. Furthermore, a merge also provides the opportunity to work on detailed individual or joint optimisations of the physical, medium access and network layers — in particular on optimisations that aim to increase the efficiency and performance of the network by exploiting the benefits of both worlds.

The rest of this paper is organised as follows. Section II outlines the differences in perspective on communication between network and physical layer simulators, illustrates the drawbacks of each view and indicates the benefits that can be achieved by merging both perspectives. Section III presents an overview of our PHY implementation, details its integration into NS3, highlights the transition from packet to signal level and provides a detailed description of the frame construction, transmission and reception processes. Following, Section IV, presents a validation of our implementation against Atheros based IEEE 802.11 wireless chipsets and offers simulation results of a vehicular network with a double-selective channel, which demonstrate the benefit of use of an accurate simulator. As the increase in simulation accuracy is offset by an increase in computational effort and memory overheads, we present a runtime analysis of our simulator in Section V and identify computational hotspots which can be targeted for optimization. Finally, we conclude our work in Section VI with a summary of our conclusions and suggestions for future work.

## II. Network Layer versus Physical Layer

In this section, we provide a brief outline of communication systems as viewed from the network and physical layer perspectives. Subsequent discussion highlights the differences in modelling such systems under each perspective as, intuitively, each view reflects different priorities and objects of study. Further, we summarise the drawbacks resulting from the existing separation of the two views in simulations, and reason on the possible benefits of establishing a common view.

### A. Network Layer Perspective

Simulation studies performed from a network perspective typically focus on aspects related to the performance and behavior of the whole network. For instance, studies that address existing medium access control issues often record the observed channel access times, the packet collision probability, fairness or scalability issues (e.g., [8]–[10]). Similarly, studies that focus on network or transport layer aspects normally evaluate metrics related to routing, dissemination or point-to-point communication, such as the number of required retransmissions, dissemination delay or the number of routing hops (e.g., [11], [12]). At this level , the entity of interest for such studies is the packet (or frame[1]), a fact which is reflected in the metrics used, many of which are measured in packets.

The adoption of the packet as the simplest unit of interest has lead to abstractions in the workings of the physical layer, where an unambiguous specification requires the use of lower level entities such as bits and signal time samples. Modern network simulators have largely adopted what can be termed as packet-level physical layer models, where the packet is considered an indivisible unit, i.e. there is no modelling of individual bits (or lower level components) in the collective whole.

The packet-level physical layer approach is reflected in the widely used NS-2 [13] simulator where, initially, the physical layer representation utilized a basic reception threshold model in order to simulate the carrier sense functionality and determine successful packet reception. In particular, a packet was only received successfully in simulations if its signal strength was above a pre-defined threshold and assuming it did not experience any collisions. This approach was found to be too disconnected from the workings of real transceivers by Chen et. al, who proposed an improved model which kept track of all incoming packets and used a signal-to-interference-noise ratio (SINR) to determine whether a packet could be received successfully [14]. Note that tuning the SINR thresholds in that model can reflect the level of sophistication and effectiveness of the receiver – with a particularly sophisticated receiver requiring a relatively low SINR to decode a packet. The model was further extended in [15] by the same authors to optionally enable packet capturing capabilities for the receiver. This facility accounts for advanced receiver technologies which allow

[1]For the purposes of this discussion the terms frame, datagram and packet are interchangeable.

synchronization (i.e. switching reception) to a new incoming packet even if the transceiver is already in the process of reception. Overall, the work of Chen et al. has been integrated in other popular network simulators, such as OMNeT++ [16], NS-3 [17] and QualNet [18].

Apart from the SINR-based reception models there exist others based on statistical bit-error rate (BER) computations. Examples of this approach can be found in the NS-3, Jist/SWANS [19] and GloMoSim [20] simulators. BER-based models use the SINR to derive a corresponding average single BER and then use this for the calculation of the final packet error rate taking into account the number of bits in the packet. In models a large packet experiences a higher error probability than a smaller one. Note that SINR-based models disregard the length of a packet in error computations; regardless of the packet size, if the SINR threshold is crossed, even for a minute amount of time, the packet is rejected. Similar to the thresholds used in the SINR-based approach, one can tune the receiver effectiveness in BER models by modifying the SINR to BER mapping, e.g., by using either analytical BER models [21] or by employing lookup tables that have been populated through empirical measurements or detailed physical layer simulations.

### B. Physical Layer Perspective

Broadly, physical layer oriented research studies [22] are primarily concerned with point-to-point link performance rather than the network wide implications of particular algorithms — this is unsurprising considering that the network level effects of a communications stack are viewed as dictated by higher layer protocol functions (e.g., MAC). At the physical level of inquiry, the metrics of interest are largely the power efficiency, that is a measure of the minimum received power required to satisfy a target BER probability, and some measure of spectral efficiency; the two quantities are frequently viewed as opposing optimisation trends and discovering an optimal trade-off point (depending on the application) is a continuing research challenge. As can be expected, much of the relevant literature in the area of physical layer and channel modeling [23] considers the signal time samples as the basic unit of interest because they allow for precise enough description of the mechanisms covering the functionality assigned to the physical layer [24], [25].

Simulation evaluation at this level, insofar as a broad characterisation of it is possible, is conducted on single-use simulators based on some signal processing framework such as IT++ [2], or, more frequently, Matlab [1], Simulink [26] or similar [27] environments. Typically, simulations consider a transmitted signal (represented by sufficient time samples) that is altered by channel effects and experiences some level of interference and noise when it reaches the receiver. Then, the time samples ultimately act as input data for the physical layer at the receiver, where the decoding process, that is the transformation to bits, takes place. Note, that during transmission and/or reception, signal processing techniques are employed to characterise and ameliorate the effects of the channel and interference or deal with some phenomenon of interest. Broadly, a comparison of the original bits at the sender and the decoded bits at the receiver provides a measurable quantity of effectiveness of competing techniques in terms of BER.

The BER result, expressed in Packet-Error Rate (PER) terms, can be used in network layer oriented research, as described previously. However, the BER to PER transformation is not usually straightforward and, perhaps more critically, both the PER and BER measures reflect particular simulation parameters such as specific packet sizes and a particular channel model. Most network simulators work around this limitation by providing a look-up table where different packet sizes correspond to particular PERs or even consider a different table per channel model. As such, interactions of packets of differing sizes characterised by different signal propagation models can lead to a prohibited growth in the dimensionality of such tables and therefore significant compromises need to be made; normally coarse packet granularity is assumed (say only packets of 300, 500 and 1000 bytes are used in the simulations) and it is assumed that all packets propagate through the same channel. Depending on the level of detail required these limitations may be overly restricting.

Moreover, physical layer simulations do not model effects at higher layers. For instance, the error detecting (or even correcting) mechanisms of an encapsulated MAC frame, or of higher layer payloads, are not directly considered. Further, in several simulation studies the bits in the frame often do not exhibit any special structure, such as, say, the one dictated by the 802.11 standard, but are, instead, distributed randomly along the frame. Generally, the prospect of studying interactions with higher layer mechanisms is limited, as creating a simulator incorporating these layers is a non-trivial task and perhaps beyond the expertise of a non-interdisciplinary researcher. So, appreciating the impact of a proposed physical layer mechanism on the whole communications stack in the context of a network is frequently not immediately possible.

### C. Problem Statement

Treating the network and physical layers as broadly abstractable entities in simulations can lead to significant drawbacks when evaluating wireless communication systems in general and vehicular networks in particular.

Notably, network level simulation studies ignore the implications of significant effects observed at the physical layer and the wireless channel. Such effects are, for instance, the impact of high relative speed between a transmitting and receiving vehicle on the communications channel or even the effect of scatterers between and around the communicating pair. Studying such effects and evaluating attempts to ameliorate them requires a particular level of detail commonly encountered only in physical layer oriented research. To appreciate the importance of the above, consider the fast-fading characteristics of the received power as well as the large root mean square delay and Doppler spreads reported by recent measurement campaigns in actual deployments [23]. In that setting, inter-carrier and inter-symbol interference can exist within a single packet, which leads to reduced communication reliability, unless the effects are catered for by appropriate signal processing mechanisms at the receiver. Such considerations, however, are not reflected in modern network level simulators. Further, since the envisioned IEEE 802.11p standard for vehicular communications derives largely from IEEE 802.11, which was not original designed for highly mobile ad-hoc networks, these issues have to be reflected in simulations especially if high precision and detail are required (for instance, in safety applications).

Physical layer simulation studies, on the other hand, do not consider the effect of mechanisms present in upper layers nor make use of the added information they could provide. As an immediate consequence, there is no direct way to evaluate how feedback from higher layers (say the MAC or routing mechanisms) may aid in choosing appropriate signal processing techniques at the physical layer. For instance, consider that in a vehicular network information on the future mobility status of a communicating neighbour (as predicted by the routing agent which receives periodic updates of other vehicles' speed and direction) could help the physical layer switch to a more suitable mode of transmission — perhaps opting for a lower transmission rate if a neighbour is deemed to be moving away, so as to obtain increased communications range and reliability. Evaluating such a mechanism directly, i.e. not through statistical abstractions, is not possible unless both the node's transmission mechanisms and the network in its entirety are accounted for in sufficient detail.

### *D. Benefit of Merging Both Perspectives*

Consolidating the physical and network layer perspectives into a common simulation framework provides two main benefits. First, it allows each perspective to consider a complete set of modeling aspects; network simulators can accommodate realistic physical layer phenomena, while physical layer studies may account for the impact of medium access mechanisms and network layer characteristics. Thereby simplifying assumptions which may impact simulation results need not be made. Second, such a merge enables new types of research enquiries involving cross-layer feedback and optimization studies, which have either been unrealisable, or difficult to perform directly. So, researchers may assess advanced physical layer techniques originating from information theory and evaluate their impact on network-wide performance.

To highlight the practical implications of the above, we consider the work of Halperin et. al [28] where they applied the concept of interference cancellation at the physical layer to a small wireless ad-hoc network testbed of ZigBee nodes in order to increase spatial reuse and reduce the effect of hidden terminals. By giving each node the ability to disambiguate and successfully receive concurrent overlapping transmissions from multiple sources, the authors were able to disable the carrier-sense mechanism of the medium access layer altogether and significantly increase, due to improved spatial reuse, the delivery rate for the median pair of links in the testbed. Evidently then, cross-layer optimization has been employed to reduce the complexity of the distributed medium access by using an advanced algorithm at the physical layer — the above work hints at the potential of studying such optimizations through simulations. Similar work, which further strengthens the case for a perspectives merge, has been undertaken by Tan et. al in [29] and Sundaresan et. al in [30]. Tan et. al studied the same concept as in [28] for IEEE 802.11b networks, leading to the development of the Carrier Counting Multiple Access (CCMA) mechanism, in which up to a limited number of overlapping transmissions are allowed before transmission requests from upper layers are blocked. A novel MAC protocol, called stream-controlled medium acces (SCMA), was also presented by Sundaresan et. al. SCMA leverages the benefits of multiple input multiple output (MIMO) links in order to increase the performance and throughput of wireless ad-hoc networks. With MIMO, data to be transmitted can, e.g., be demultiplexed into several streams with each transmitted out of a different antenna with equal power, at the same frequency, modulation format, and time slot. A receiver can then, under some conditions, distinguish the different streams when either no interference is present or as long as the total number of incoming streams (even if they originate from several different transmitters) is smaller than the number of receiving antennas. If this condition is not met, the receiver becomes overloaded and unable to suppress the interference from other nodes. In order to exploit the potential of MIMO efficiently, the SCMA protocol determines the maximum number of usable streams for each packet transmission so as to enable successful suppression of interference at the receiver. Intuitively, the performance gain depends on the amount of correlation between the receiving antennas and reaches its peak if the streams at the antennas are not correlated at all. Since the authors used a traditional packet level network simulator to evaluate SCMA, they could only employ a simple model for the physical layer MIMO characteristics that assumes a minimum correlation level between the streams. With a network simulator that integrates both physical and network layer details, they could have assessed the performance more accurately.

Apart from enabling interference cancellation studies, however, fusing both perspectives can also enable the accurate study of simple and advanced network coding techniques, where the capacity of a network is increased by coding multiple packets into a single transmission, and the coding itself may be performed at the physical, link or network layers [31].

## III. IMPLEMENTATION

In this work, we propose the integration of a detailed physical layer simulator into NS-3 in order to improve the accuracy of the IEEE 802.11 physical layer and the underlying channel models. The integration requires no changes in the upper layers, such as the MAC, and can be used as a "drop-in" alternative to existing PHY implementations. Figure 1 provides an overview of our modular implementation which reflects the IEEE 802.11 physical layer mechanisms used in OFDM-based communication, as defined in the older a and g amendments as well the latest p draft for wireless access in vehicular networks. The legacy direct sequence spread spectrum (DSSS) modes and the infrared communication provisions of the standard are not considered in our work. The implementation makes

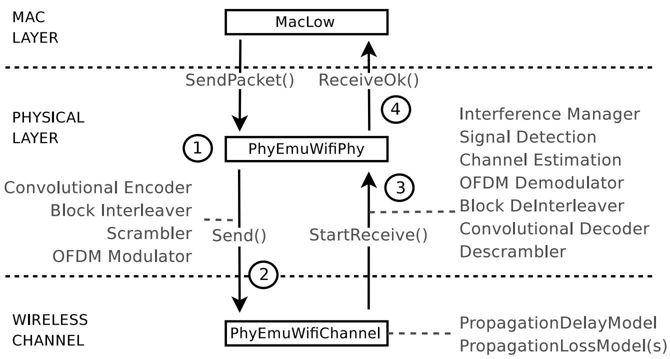



Fig. 1. Architecture of the proposed physical layer simulation within NS-3. (1) The payload is expressed as a sequence of bits. (2) The bits are modulated into a sequence of complex time domain samples, on which wireless channel models will operate. (3) After the application of channel effects the time samples are processed at the receiver and demodulated into bit representation. (4) The received bits are compared to the transmitted ones to determine whether the frame was received successfully.

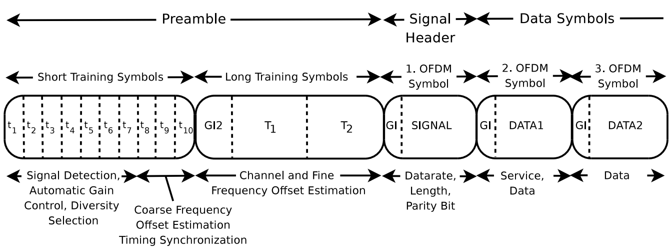



Fig. 2. PPDU frame format of an OFDM-based IEEE 802.11 PHY. Note that the service field in the second OFDM symbol is part of the header.

extensive use of the open source IT++ library [2], which provides several convenient data structures and functions for signal processing and channel modeling techniques. Note that IT++ has been widely used in physical layer research and is actively maintained.

An IEEE 802.11 frame for OFDM-based communications can be broadly distinguished into three sections, namely a preamble, a signal header and a data unit section. Figure 2 shows the overall structure of such a frame as well as the purpose of each section. The preamble consists of a series of repeating time sample sequences, which are identical for every frame transmitted regardless of the mode of transmission. These sequences are, specifically, ten repetitions of short and two of long training symbols, which can be used by the receiver for signal detection, automatic gain control, diversity selection, timing synchronization as well as channel and frequency offset estimation. After the preamble, there exists the signal header, termed the SIGNAL in the standard, which contains information about the length of the data unit section, the modulation and coding scheme used and, further, includes a parity bit to support basic error detection. The SIGNAL fits in one OFDM symbol and contains most of the frame's header information, apart from a 16-bit service field which is included in the first OFDM symbol of the data section. Finally, the data section is the last distinct part of the frame, can be several OFDM symbols in length and contains the payload to be transmitted. In the following discussion, we elaborate on the frame construction process and the modelling of wireless channel effects and receiver functionality.

### *A. Frame Construction*

During the frame construction process the physical layer simulator closely mimics the behavior of an IEEE 802.11 compliant transceiver. Initially, as shown in Figure 1, the simulator accepts transmission requests from the MAC layer. In NS-3 simulations, frames are usually treated as dummy objects that contain header information from different protocols but no actual payload. So, in order to obtain a bit level representation the physical layer simulator generates a random data bit sequence with a size equal to the length specified in the header of the frame object[2]. Once the bit payload has been defined the simulator follows the procedures outlined in Section 17 of the IEEE 802.11 standard, which describe the OFDM mode of operation for transceivers operating in the 5GHz range. First, the data bits are transformed by the *Scrambler* module which prevents the appearance of long sequences of 0s or 1s, then the *Convolutional Encoder* module adds redundancy to enable error correction and, finally, the *Block Interleaver* module ensures that long runs of low reliability bits are avoided. Importantly, the *Block Interleaver* divides the bitstream into equally sized blocks, each of which can fit into a sigle OFDM symbol. Then, the *OFDM Modulator* modulates the bits of each block using either phase-shift keying (BPSK or QPSK) or quadrature amplitude modulation (16-QAM or 64-QAM), inserts pilot symbols in four of the 52 sub-carriers to support channel tracking in the receiver and performs the final OFDM modulation per block — the end product of the above transformations is a sequence of complex time domain samples. Figure 3 schematically depicts the processes involved. Note that a similar process is applied to the signal header of the frame with the special condition that no bit scrambling is done and a particular combination of modulation and coding rate (BPSK,1/2) is applied regardless of the transmission mode.

To ensure that frame construction adheres to the specification of the standard, we have carefully verified that the time samples generated by the simulator match the example provided in Annex G of the IEEE 802.11 standard [24].

### *B. Channel Effects*

Intuitively, the wireless channel is modelled as a collection of plug-able components each of which represents a particular channel effect. Once the frame has been generated, the sequence of complex time domain samples is passed on to the wireless channel module which allows chaining several propagation loss models such that the output of one serves as input for the next.

The implementation includes basic pathloss models such as Friis, Two-Ray Ground and other LogDistance implementations as well as large- and small-scale fading models and multi-tap channels, which reflect the great collection available in the IT++ library. Additionally, we have implemented the

[2]In NS-3 it is also possible for higher layers to specify a particular payload. In this case the simulator considers the bit representation of the actual data.

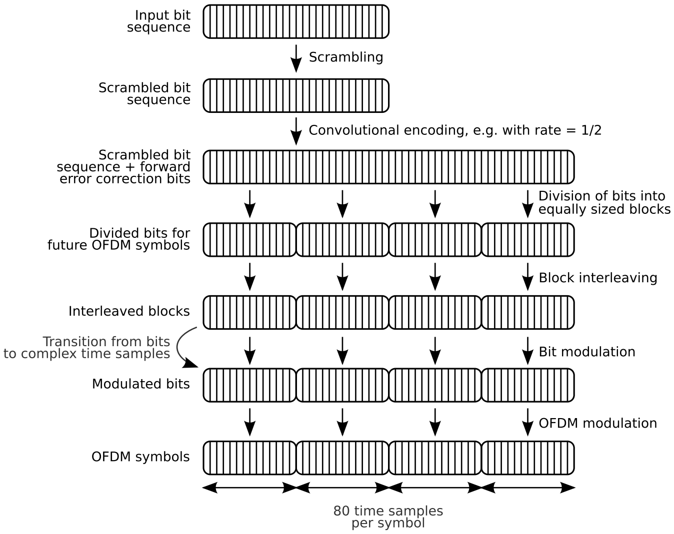



Fig. 3. Transformation of a bit sequence into complex time domain samples during the construction of an IEEE 802.11 frame for OFDM-based communication.

vehicular channel models described in the work by Acosta-Marum and Ingram [32], taking into account the errata noted in [33]. The channel models therein adopt the tapped-delay line model where each tap is characterised by a Rician or Rayleigh Fading process and a Doppler power spectral density. Due to the sample-level granularity enabled by the simulator, the implementation of the channel models described in [32] is complete without further abstractions or approximations.

Note that the shadowing effect of neighbouring vehicles is not explicitly accounted for in currently implemented channel models, i.e. neighbouring vehicles do not affect signal propagation. Shadowing, in the statistical sense, can be included in simulations via a log-normal shadowing channel component which does not take into account the number of neighbours present along the transmission path. Nonetheless, a channel component that calculates shadowing by vehicles is certainly a prospect for future work as the simulator provides all necessary information for its implementation, that is the neighbouring vehicles' location and overall spatial characteristics. Similarly, there is presently no consideration of shadowing from specific environment geometry, such as buildings or other structures.

In principle, modelling signal-altering effects at the transmitter/receiver front-end or the channel is a continuous and detailed process which can involve several degrees of abstraction. In the current implementation of the simulator we assume an ideal front-end so effects such as quantization by the analog-to-digital conversion, automatic gain control imperfections, phase-noise, and receiver non-linearities are not considered. Moreover, we assume a single omnidirectional ideal antenna present in every transceiver. The above and other important effects of physical phenomena may be included in future versions as research interest dictates.

Overall, as we have indicated, modeling of time and frequency-selective channels can be achieved in the framework. In general, different receivers experience different channel responses in simulations. Specifically, whenever a transmission occurs, each receiver has a separate channel effects "chain" associated with it; this "chain" (or collection of channel effects) processes the transmitted time samples and delivers the output to the receiver. Typically then, and depending on the channel model in use, the received time samples from a particular signal source differ among receivers.

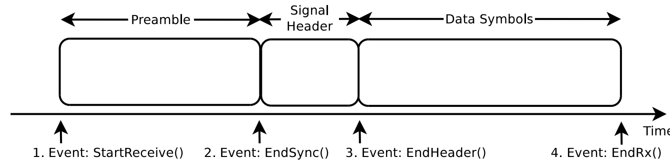



Fig. 4. The four events of the reception process: when the first sample of the frame arrives, the frame is added to the Interference Manager ($1^{st}$ event). After the preamble transmission time has elapsed ($2^{nd}$ event), the simulator checks whether signal detection and synchronization was successful. If so, the signal header decoding is performed ($3^{rd}$ event). If that is also successful, an attempt is made to decode the data symbols according to the detected frame length and data rate ($4^{th}$ event).

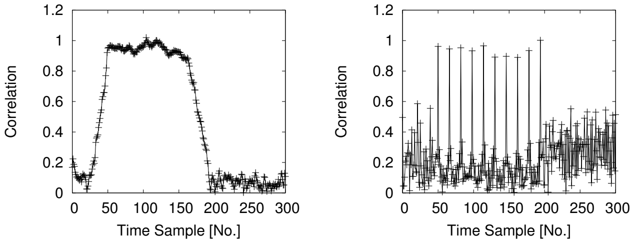
### *C. Frame Reception*

After all channel effects have been applied, the sequence of complex time samples is passed up to the physical layer of a receiving node (cf. Figure 1). There, the overall reception process can be distinguished into three stages, which are implemented through four events in NS-3, as illustrated in Figure 4.

First, the *Interference Manager* adds the sequence of complex time samples to its internal list of incoming frames, regardless of the frame's power level. The task of this mechanism is to keep track of all incoming, and possibly overlapping, transmissions so as to precisely account for the effect of cumulative interference at the receiver. This also includes white Gaussian thermal noise generated by the transceiver itself.

Second, at the end of the preamble's reception, the receiver tries to perform signal detection and synchronization, i.e. "lock-on" to the correct time samples in the incoming frame. To achieve this, signal detection techniques typically exploit the fact that there is a repeating pattern (the training sequences) in the preamble. Our particular implementation uses the correlation techniques described in [34]. Figure 5 shows the behaviour of two different signal detection mechanisms described in that work — therein, high correlation values indicate great confidence that the received sample is a part of the preamble. In the illustrated example both detection methods (auto-correlation and correlation with the known sequence) identify an incoming frame successfully at around the $50^{th}$ time sample. In addition to signal detection, the receiver performs coarse and fine frequency offset and channel estimation according to [35].

If detection and synchronization do succeed, then the third stage of the reception process, that is signal header decoding, occurs. This stage involves OFDM demodulation, de-interleaving and convolutional decoding of the time samples (either with soft or hard decisions) to derive a bitstream representation of the signal header. If the parity bit validates




(a) Auto Correlation method (b) Known Values method

Fig. 5. Two different signal detection methods are shown: auto-correlation and correlation with known time samples. Both declare confidence (correlation approx. 1.0) that an incoming frame starts around the 50$^{\text{th}}$ time sample.

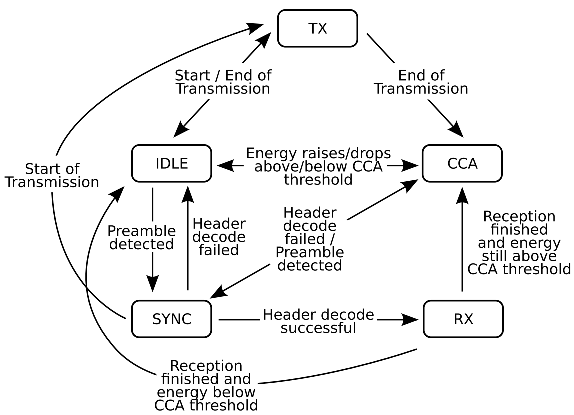



Fig. 6. The state machine of the physical layer simulator distinguishes between *Transmit*, *Idle*, *Busy*, *Sync* and *Rx* states.

the decoded header, the frame decoding process proceeds to its final stage which considers time samples up until the end of the overall frame reception (cf. Figure 4). Once more, a decision on whether all data symbols can be decoded successfully is made at the end of this stage, however, this time the outcome is decided by comparing the transmitted bit sequence with the decoded one. It should be emphasised that the simulator uses as input the cumulative signals of overlapping frames, i.e. considers interference, as well as thermal noise in every stage. Also note that distinguishing the reception process in several discrete events is necessary if the receiver is to reflect a real system. It is particularly important for decisions to be taken at the end of each stage, otherwise possible interference from other sources happening after the decision point would be ignored.

### *D. Physical Layer State Machine*

The physical layer simulator and its behaviour is implemented as a state machine with five possible states as illustrated in Figure 6. The states are *Idle*, *Busy*, *Transmit*, *Sync* and *Rx*. The *Idle* state is maintained if no signal header is successfully decoded and as long as the energy detected at the receiver stays below the Clear Channel Assessment (CCA) threshold (according to IEEE 802.11 [24] Section 17.3.10.5). As soon as the detected energy rises above the CCA threshold (and assuming a signal header has not been decoded successfully), the physical layer is marked *Busy* and

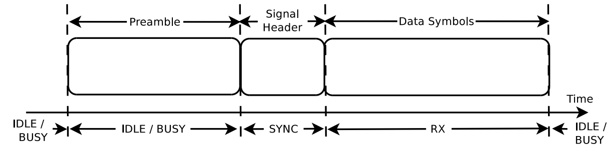



Fig. 7. The stages of the reception process and their corresponding physical layer states. During preamble detection, the physical layer is still considered to be *Idle*. If signal detection is successful, the physical layer first switches to the *Sync* state and then, if the signal header is decoded successfully, to the *Rx* state.

the MAC layer is notified to not request any new transmissions and to support the Carrier Sense Multiple Access mechanism (CSMA). To facilitate the above, our physical layer checks the signal strength of surrounding transmissions both before switching to an *Idle* state, as well as when a new frame arrives during an *Idle* period. In those cases, it determines whether the energy is already above or is going to exceed the CCA threshold in the near future by calculating the cumulative signal strength over consecutive blocks of 80 time samples. The detection of a preamble leads to the *Sync* state, which is eventually followed by the *Rx* state if the corresponding signal header is decoded successfully. When the *Rx* state is active, the MAC is notified again to block its own transmission requests for the duration of the reception (Virtual Carrier Sensing). On the other hand, the transmission of a frame sets the physical layer in the *Transmit* state. Figure 7 depicts a typical case of the reception of a single frame and the corresponding states of the physical layer.

With some small additions to the above state machine, the physical layer simulator can also support the frame capture capabilities of modern transceiver chipsets, i.e. can simulate switching or "locking-on" to another preamble than the one being currently processed, if the new incoming signal is stronger. More precisely, this can be achieved by adding two state transitions from *Rx* to *Sync* and from *Sync* to *Sync*. We aim to include such provisions in the near future.

## IV. Validation & Evaluation

In the following, we present the validation of our physical layer implementation against a commercial IEEE 802.11 chipset and, further, present a simulation-based evaluation of the communication performance of the simulator under non-fading, frequency-flat Rayleigh fading and double-selective radio propagation conditions. Further, we make a distinction between the case where no interference from other transmitting nodes is present and name this the *single node case*, as opposed to the *multiple nodes case*, where interference does exist.

For validation purposes we used the network emulator testbed of Carnegie Mellon University (CMU) as described in [3], which allowed us to measure the reception performance of Atheros AR5112 chips under controlled non-fading and Rayleigh fading conditions[3]. Since the testbed supports only

[3]Note that despite the use of a controlled channel emulator which interconnects real hardware, experiments focusing on detailed packet-level reception performance are difficult to replicate precisely on such a platform, although overall trends can be identified.

TABLE I
MAJOR TIMING- AND FREQUENCY-RELATED DIFFERENCES BETWEEN THE G AND P AMENDMENTS OF IEEE 802.11.

| **Parameter** | **IEEE 802.11g** | **IEEE 802.11p** |
|---|---|---|
| Channel bandwidth | 20 MHz | 10 MHz |
| Channel frequency | 2.4 GHz | 5.9 GHz |
| Symbol duration | 4 $\mu$s | 8 $\mu$s |
| Guard interval | 0.8 $\mu$s | 1.6 $\mu$s |

the 2.4 GHz frequency band, the validation was performed by adjusting the emulator to conform to IEEE 802.11g specification having a 20 MHz channel bandwidth. Although we would have preferred to validate against IEEE 802.11p devices at 5.9 GHz (cf. Table I for the major timing- and frequency-related differences) this setup still allowed us to confirm the validity of our physical layer algorithms, as both systems are OFDM-based and very closely related in function.

Further, we performed simulations using our IEEE 802.11p simulator in a number of different configuration in order to determine the differences between the physical layer models that have been used previously in network layer research (cf. Section II-A) and in particular contrast the BER-based model implemented in NS-3, with our detailed physical layer implementation. In this instance, we classify the scenarios w.r.t. the applied propagation model to highlight in which situations the traditional packet level based approach is not accurate enough.

We performed 10 experiments on the network emulator testbed, each with different random seeds where applicable and similarly performed 40 trials on the simulator. In the following discussion and presentation of results we consider the mean of the observed values and present the corresponding 95% confidence interval in each case as error bars in the relevant figure. The simulations were configured according to the values listed in Table II.

### *A. Single Node Case*

The first validation experiment was performed using a static scenario in which a single node sent 1000 packets in a row and another node tried to decode them. To achieve different SNR conditions, we ran each experiment with different pathloss configurations, i.e. instead of adjusting the distance between sender and receiver we adjusted the channel configuration. For the experiments on the emulator testbed, we disabled any rate control algorithms available in the driver and allowed the receiving node to adjust to new channel conditions by introducing a warm-up phase of 200 packet transmissions.

Figure 8 shows the frame reception ratio – as observed in our experiments on the testbed as well as in simulations with our new implementation – for different data rates. As can be seen, the reception curves show not only similar slopes, but also start to rise at very similar SNR values which, at lower data rates, are at maximum 1 dB apart. Only at the highest data rate did our simulator results diverge significantly from the testbed and, after having discussed the disparity with the team at CMU we can attribute this to the presence of too much

TABLE II
CONFIGURATION PARAMETERS USED FOR THE VALIDATION AND EVALUATION EXPERIMENTS.

| **MAC layer configuration** | |
|---|---|
| Frame size | 500 bytes |
| Transmission rate | 10 Hz |
| CCA threshold | −95 dBm |
| **PHY layer configuration** | |
| Transmission power | 20 dBm |
| Maximum oscillator offset | 10 ppm, according to 17.3.9.4 of [24] |
| Channel estimator | Linear interpolation between pilot sub-carriers with frequency offset estimation [35] |
| Viterbi decoder | Hard decision decoding |
| **Channel configuration** | |
| Thermal noise | −99 dBm |
| Non-fading conditions | Static pathloss, 90-130 dB |
| Freq.-flat Rayleigh-fading | Jake's Doppler spectrum |
| Double-selective channel | V2V Expressway Oncoming, V2V Urban Canyon Oncoming [32] |
| Relative vehicle speeds | 10 m/sec, 26 m/sec, 52 m/sec |

phase noise in the analog to digital conversion component of their channel emulator.

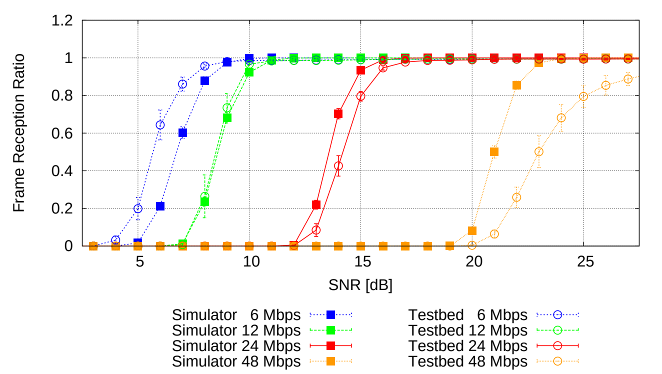



Fig. 8. Frame reception ratio w.r.t. SNR of the new physical layer in NS-3 compared to the results obtained with real chipsets. At lower data rates, the difference is at most 1 dB. At high data rates the simulator is significantly better. The configured channel reflects only a static pathloss, no Rayleigh fading was applied. Packet size was set to 500 bytes.

To compound the results of the previous comparison, which showed very similar reception behavior between the two systems, we repeated the experiments and simulations of that scenario with a Rayleigh-fading effect added to the pathloss configuration. To be able to compute the Rayleigh channel coefficients in real-time, the CMU network emulator testbed implements the algorithm presented in [36] and uses its output to set the fading table in their FPGA-based channel emulator[4]. Based on this approach, different fading speeds (due to relative speeds) are then realized by updating the fading table at the corresponding rate. However, due to technical limitations of the testbed, it is not possible to check whether the configured

[4]The algorithm is not implemented on the FPGA itself. Instead, a dedicated machine is used to perform the calculation and update the entries in the fading table of the FPGA.

fading speed is actually achieved or not, i.e. whether the fading table is updated often enough. Indeed, we could not observe any difference in the reception performance when configuring different relative speeds, although the CMU emulator offers support of relative speeds up to 50 m/sec. These issues did not exist in our simulations, since they do not have any real-time constraints.

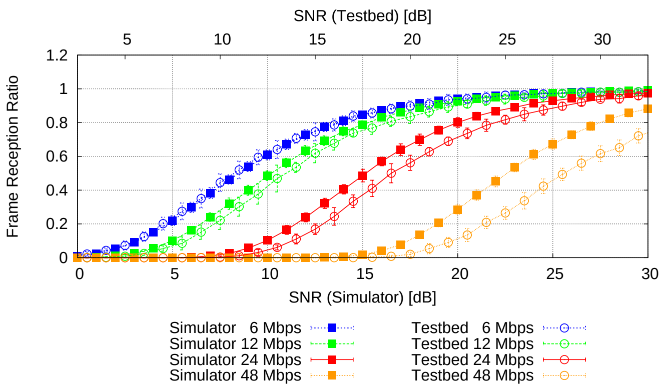



Fig. 9. Frame reception ratio w.r.t. SNR of the new physical layer in NS-3 compared to the results obtained with real chipsets. At lower data rates, the difference is at most 3.5 dB. At high data rates the simulator is significantly better. The configured channel reflects a static pathloss plus flat Rayleigh fading with classical Jake's Doppler spectrum and relative speed of 10 m/sec. Packet size was set to 500 bytes.

The frame reception ratios of the conducted simulations and testbed experiments with a Rayleigh fading channel are illustrated in Figure 9. For these the relative speed used was set to 10 m/sec. Similarly to the non-fading scenario, the slopes of the observed performance curves are again very similar, but this time, the offset varies between 3-5 dB throughout all evaluated data rates and the performance of the Atheros AR5112 chipset is significantly worse than the one observed in our simulator (note the use of two different x-axes in this figure). We believe that this discrepancy can be attributed to the channel estimation algorithms being used in the AR5112 chipset, which are different to our own. We further believe that more contemporary chipsets would lead to better results. It should also be noted, however, that our intention is not to reflect the performance of a particular chipset; instead, we are interested in observing whether our current implementation is comparable to available chipsets and therefore realistic. Based on the obtained results, we felt justified in making such a claim and can consider our implementation as valid foundation for future proposals and improvements, e.g., for better channel estimation and equalizing techniques in vehicular environments [37].

To illustrate the benefits of the increased accuracy available with the new physical layer implementation, we conducted additional simulations contrasting its use with a more traditional approach already implemented in NS-3. Similar to the setup used in our validation efforts, we considered a scenario with one transmitting and one receiving node. However, at this time we configured the physical layer to reflect the IEEE 802.11p standard. As our implementation is a drop-in replacement, we could easily switch between the packet-level implementation of NS-3, hereon referred to as *YansWifi*, and

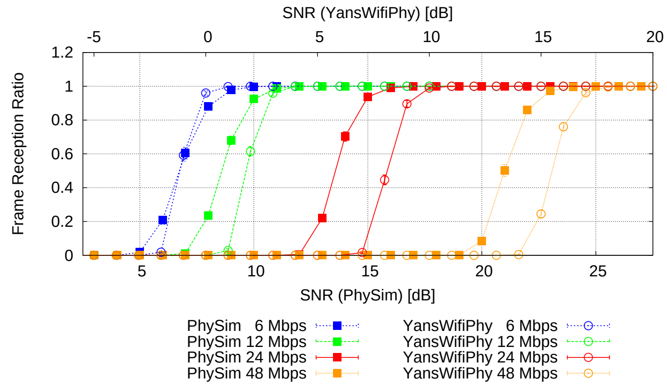



Fig. 10. Comparison of the frame reception ratio of the new physical layer (PhySim) in NS-3 compared to the existing physical layer implementation (YansWifi) in NS-3 with 500 byte packets and non-fading channel conditions. The YansWifi implementation provides results that are at least by 8 dB more optimistic than the ones obtained with PhySim.

our new implementation, which we term *PhySim*.

In Figure 10 the observed frame reception ratios of both implementations is plotted w.r.t. the SNR that can be derived after the pathloss effect has been applied. Note that the YansWifi model results have been plotted against an x-axis with an offset of 8 dB compared to that of the physical layer implementation results so as to highlight the similarities in the shape of the curves. It is clear though that the existing BER-based physical layer model in NS-3 generates more optimistic results compared to the new implementation, which, however, does not imply that one modeling approach is better than the other. Note that the slopes of the curves in this case show very similar characteristics — in fact they could coincide substantially if a linear offset was introduced for each data rate.

When simulating the scenario with a Rayleigh-fading channel, a similar conclusion can be drawn – at least initially. As illustrated in Figure 11, the observed frame reception curves for the data rates of 12, 24 and 48 Mbps follow very similar slopes as well and are separated by a linear offset of 4-5 dB. However, if a data rate of 6 Mbps or different fading speeds are considered – which are not modeled in a packet-level simulator – the slopes tend to divert from each other, cf. Figure 12. To summarize our observations, we can state that packet-level simulators show inaccuracies when a Rayleigh-fading channel is modeled and cannot account for different relative speeds between a transmitter and a receiver.

As previous measurement campaigns have shown [23], [32], the vehicle-to-vehicle (V2V) and roadside-to-vehicle (R2V) channel is different from a flat Rayleigh-fading channel. In particular, V2V has been shown to be time- and frequency-selective, i.e. exhibits fading over time and w.r.t. the frequency. As a result, the impulse-responses of neighboring OFDM sub-carriers are not necessarily correlated and the four pilot sub-carriers may not be sufficient to estimate the channel for all sub-carriers correctly. To account for such a channel, we have implemented all six empirical models presented in [32], which are based on vehicle-to-vehicle and roadside-to-vehicle measurements for expressway, urban canyon and suburban

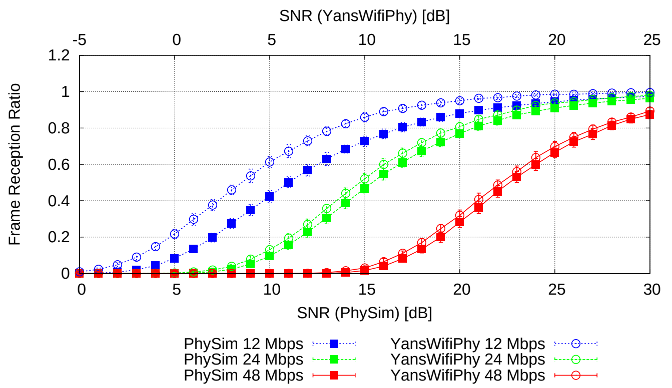



Fig. 11. Comparison of the frame reception ratio of the new physical layer (PhySim) in NS-3 compared to the existing physical layer implementation (YansWifi) with 500 byte packets and Rayleigh-fading channel conditions.

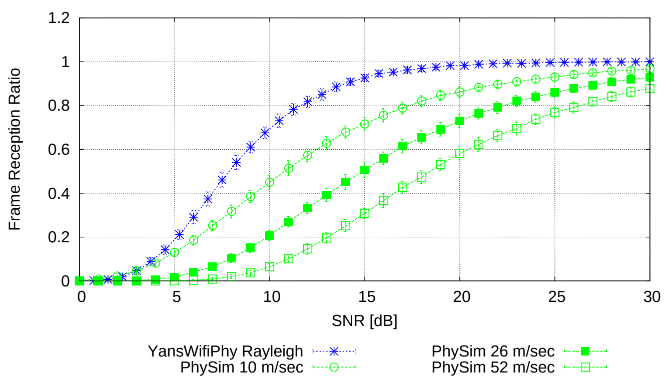



Fig. 12. Comparison of the frame reception ratio of the new physical layer (PhySim) compared to the existing physical layer implementation (YansWifi) in NS-3 with 500 byte packets, Rayleigh-fading channel conditions and data rate of 6 Mbps.

scenarios.

Figure 13 and 14 show the observed frame reception ratios with respect to the SNR for the expressway oncoming and the urban canyon oncoming channel models. As can be seen in Figure 13, our reference receiver is not able to estimate the V2V expressway oncoming channel sufficiently enough, even at very high SNRs up to 30 dB. And as with the Rayleigh channel, a higher relative speed leads to a lower reception probability. On the contrary, the reception probability shown in Figure 14 is much higher and even better than in the Rayleigh case. Indeed, the curves gradient is significantly steeper as with a Rayleigh channel configuration. Furthermore, the relative speed does not affect the reception probability as much as in the expressway scenario.

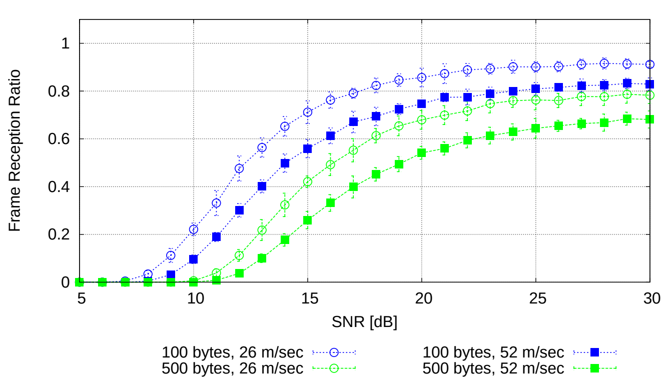



Fig. 13. Observed frame reception ratios w.r.t. SNR for the V2V expressway oncoming scenario, 6 Mbps data rate, 26 and 52 m/sec relative speed and different frame sizes.

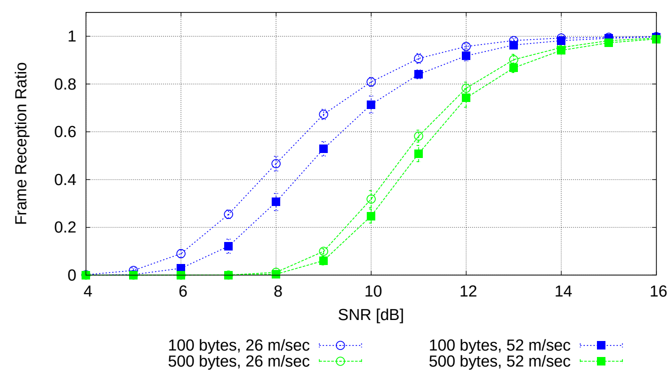



Fig. 14. Observed frame reception ratios w.r.t. SNR for the V2V urban canyon oncoming scenario, 6 Mbps data rate, 26 and 52 m/sec relative speed and different frame sizes.

## *B. Multiple Nodes Case*

The proposed detailed physical layer implementation also operates in multi-node scenarios as it is designed to be a complete replacement for the traditional, simplified NS-3 model. To explore this aspect of the simulator we have conducted further trials in a simple, but not trivial, setup of a vehicular network.

We set up a simple topology in which non-moving vehicles were placed along a straight line where inter-vehicle spacing was set to 10 m. For the radio propagation model, we considered a simple ThreeLogDistance model with pathloss exponents of 1.9, 3.8 and 3.8 for distances up to 200 m, 500 m or longer respectively. Each vehicle was configured to generate UDP packets periodically at a rate of 10 packets/sec, using a size of 400 bytes and a data rate of 6 Mbps. The simulation time for the scenario was defined to be 10 seconds. The objective of this study was to verify that the PHY-MAC interaction, i.e. the physical layer state machine and the CSMA mechanism, would work as expected.

Figure 15 shows the observed frame reception performance with respect to the distance between sender and receiver and for two different scenarios; first assuming the presence of 100 and then 200 nodes. To limit the impact of bordering effects we monitored the reception performance only for transmissions initiated by nodes in the center of the scenario area. Note that although the number of nodes differs in the two scenarios, the vehicle density is equal in both cases, that is 100 vehicles/km. Due to the node's positioning and the method of collecting results, the 100 nodes configuration can be considered as a scenario with no hidden terminals, whilst the 200 nodes setup as one which exhibits hidden terminal stations – at least with respect to the nodes located in the center. As expected, the hidden terminal impact was observed to be significant and led

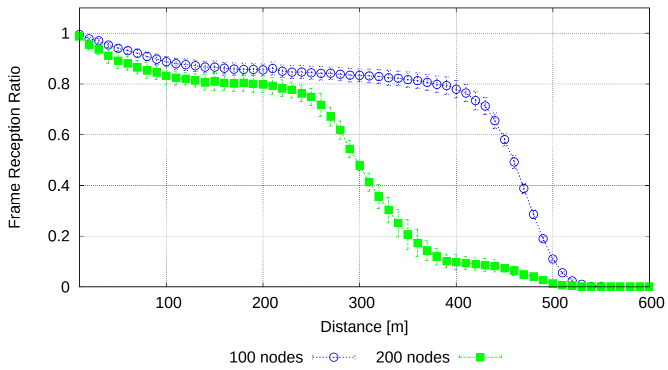



Fig. 15. Observed frame reception ratios w.r.t. the distance between sender and receiver for a network scenario with 100 and 200 nodes. The nodes are placed statically along a straight line and transmit 10 packets/sec, with a packet size of 400 bytes.

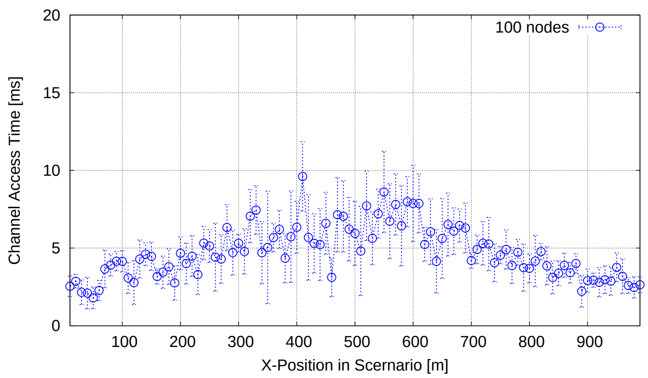



Fig. 16. Observed channel access times w.r.t. the x-position for a network scenario with 100 nodes placed along a straight line.

to a steep drop of the reception probability at a distance as short as half the transmission range.

In Figure 16, we plot the observed channel access times for all nodes in the 100 nodes scenario. Again, as one would expect, nodes in the center of the scenario experienced higher channel congestion and thus had to wait slightly longer (on average) before gaining channel access. Also noteworthy is the higher variability in channel access times, denoted by the higher confidence intervals, experienced by nodes located in the middle of the vehicular queue.

## V. Runtime Optimization

Enabling detailed simulations with the physical layer simulator increases simulation requirements both in terms of processing time and memory capacity. This section elucidates the added requirements and offers some analysis on the simulator's runtime performance.

### A. Profiling

The consideration of time samples in simulations, intuitively results in increased storage requirements as a time sample sequence need be stored per frame. At a minimum the storage requirement for a frame containing a payload of a single OFDM symbol would be 480 complex valued elements (320 elements for the preamble, 80 for the header and 80 for the payload), which in turn translates to 3840 bytes of memory space, assuming a complex value is represented by 8 bytes. In contrast, for the default NS-3 representation the minimum storage space required for a frame is 36 bytes assuming no payload and minimal 802.11 MAC headers. Overall, the added memory overhead is necessary and can be mostly attributed to the storing of time samples for each frame.

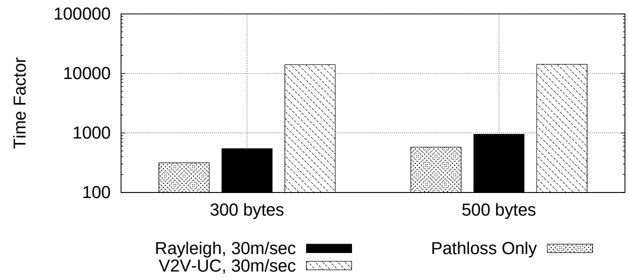



Fig. 17. Computational overhead of the physical layer simulator when using increasingly sophisticated channel models (static pathloss - Rayleigh - V2V-UC oncoming). The time factor indicates how many times longer simulation takes compared to a baseline simulation performed with vanilla NS-3 using a simple static pathloss model.

To gain insight into the computational costs of the new functionality, we have set up NS-3 simulation scenarios and noted the real running time when using the traditional and detailed physical layer implementations. Specifically, we assume two nodes in close proximity, equipped with 6 Mbps transceivers. At the beginning of the simulation, one node offers UDP broadcasts with a frequency of 10 packets/sec. The exchanges last for 12 seconds of simulation time, after which the simulation ends and we record the real elapsed time, specifically the "user" statistic of the GNU time command. In the vanilla NS-3 trials the channel model used is constant pathloss while in the case of the detailed physical layer we employ increasingly sophisticated channel models, namely a constant pathloss channel, frequency-flat Rayleigh fading and, finally, the V2V Urban Canyon-Oncoming vehicular channel model as described in [32]. Note that the constant pathloss models, both in the case of the traditional and the detailed PHY implementation, reduce the energy of the packet by a fixed factor. Two packet sizes are considered in separate simulation runs, specifically 300 and 500 bytes.

Figure 17 depicts the slowdown factor involved when employing the detailed physical layer model compared to the traditional NS-3 model, which acts as the baseline. As an example, a factor of two would denote that double the computational effort (or double the time) compared to the baseline would be required to produce results. In these simulations, the additional effort required ranged between a factor of 300 and 14,000 — generally, the computational requirements increase as greater packet sizes are used and more sophisticated channel models are employed.

### B. Optimizations

In previous work [7] we have hinted at possible optimisations to the channel models and physical layer operations in

TABLE III
THE MOST COMPUTATIONALLY INTENSIVE CODE PATHS IN THE SIMULATOR WHEN USING A RAYLEIGH CHANNEL

| **Code Path** | **Processing Time%** |
|---|---|
| Channel Model (Rayleigh) | 47.4% |
| Frame Decoding | 13.5% |
| Frame Encoding | 12.6% |
| Interference Manager | 6.9% |
| Signal Detection | 6.6% |

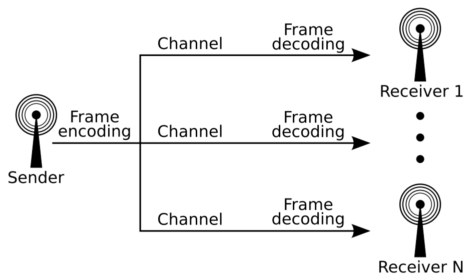



Fig. 18. Illustration of potential optimizations in the physical layer simulation: first, it is possible to parallelize the signal processing algorithms within the frame encoding, the channel model and the frame decoding processes. In addition, it is possible to parallelize the computation of channel effects due to multiple receivers since the computations are independent. With minor changes to the time-discrete event-based internals of the NS-3 simulator, it is also possible to exploit the lookahead between different reception events, which are separated only slightly in time.

view of improving operations, which included the usage of lookup tables and parallelisation of operations where applicable as well as using optimised math libraries. Here, after having performed code instrumentation only w.r.t. optimised math libraries we can elaborate further on which code paths the processing load is most significant. Table III shows five major areas of computational effort during simulations with a frequency-flat Rayleigh fading channel. The simulation parameters and setup are identical to the ones used previously.

As evinced by the above, the greatest overhead on simulation time is effected by the channel model, which accounts for almost half of the total processing time. Then, frame decoding follows, which caters for demodulation, channel estimation and correction as well as all other processes necessary to transform the received time samples into bits. Frame encoding is next with similar but somewhat fewer processing requirements. Finally, there is the interference manager, which keeps track of interference levels around a node, and the signal detection mechanism, which involves synchronisation as well as channel estimation using the preamble.

In order to speedup the simulation time of the physical layer implementation, we have evaluated the usage of general purpose graphics processing units to parallelize the individual code paths. Since most computational effort is allocated to the simulation of sophisticated channel effects, we first targeted our sequential Rayleigh fading model for optimization. To enable parallel execution we used the OpenCL framework [38], which ensures that the optimised version can be executed on any multi-core processor or multi-core graphics processing unit that supports this open API. In our subsequent runtime tests, we used an ATI RadeonHD 5870 device with 1600 parallel processing units/cores in total and observed a speedup factor of 4 for the computation time of the Rayleigh channel. We have also developed a parallelized version of the convolutional decoder which when used at the frame decoding process reduces computation time by a factor of 3 to 6. Apart from the parallelization of the individual code paths, it is also possible to compute the channel effects of multiple receivers in parallel, as illustrated in Figure 18. Compared to the sequential computation of the channel effects, parallel computation provides linear speed increases w.r.t. the number of receivers present as long as there are enough cores available to the OpenCL framework. In the future, we plan to extend this effort to the V2V and R2V channel models that consist of 4-8 independent taps and document the cumulative effect of these optimizations.

## VI. CONCLUSIONS

Communication technologies have been traditionally studied from two different perspectives. On one hand, experts conducting physical layer and channel modeling research focus on the performance of single communications links and use very detailed models involving individual bits and complex time samples. On the other hand, researchers focusing on the network and higher layers of the communications stack are interested in the overall network performance and typically abstract the lower layer details, considering the packet as an indivisible unit that is either received as a whole or not at all. However, this separation of concerns, i.e. abstracting without having recourse to detailed models, can lead to unconsidered or inappropriately modelled effects in network layer studies and to imprecise assumptions in physical layer research.

In this paper, we have provided a tutorial with respect to these two perspectives in view of establishing a mutual understanding in both research communities. We introduced both perspectives, outlined the issues arising from their different approaches to communications and motivated the benefits of their possible merge. We then presented our approach on how to bridge the gap, in particular through the integration of a physical layer simulator into the NS-3 network simulator. Further, we presented results of an extensive validation and simulation work in which we compared our implementation against commercial IEEE 802.11 chipsets from Atheros and the traditional packet-level physical layer models in NS-3. The results indicate that our reference implementation performs as well as the Atheros chipset and that existing, traditional packet-level models are not able to provide the same level of accuracy. In addition, by integrating empirical vehicular channel models into our simulation framework we showed that our approach allows the direct integration of models in existing physical layer and channel modeling studies.

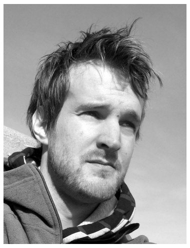
Finally, as the improved accuracy comes at the cost of increased computational effort, we performed a runtime analysis of our implementation, which showed that most of the computational overhead is due to the complex channel models. In the future, we, therefore, aim to optimise the computation of such channel models by making use of parallelisation where possible or by utilising specialised hardware such as that available in general purpose graphics processing units. Our implementation is available online [39] and enables accurate cross-layer PHY/MAC/NET simulation studies for vehicular communication networks by the research community.

## ACKNOWLEDGMENT

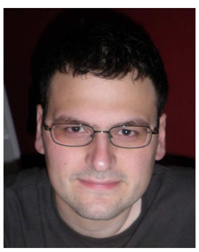
Jens Mittag acknowledges the support of the Ministry of Science, Research and the Arts of Baden-Württemberg (Az: Zu 33-827.377/19,20), the Klaus Tschira Stiftung, the INIT GmbH and the PTV AG for the research group on Traffic Telematics for the research group on Traffic Telematics. E. G. Ström acknowledges the financial support from the Swedish Foundation for Strategic Research (project SM07-0025) and from the Area of Advance in Transportation at Chalmers University of Technology; he also wishes to acknowledge the fruitful discussions in working group SIG-C of the COST Action 2100–Pervasive Mobile & Ambient Wireless Communications.

## REFERENCES

[1] "MATLAB high-level language and interactive environment," http://www.mathworks.com/products/matlab/.
[2] "The IT++ library," http://sourceforge.net/apps/wordpress/itpp/.
[3] G. Judd and P. Steenkiste, "Repeatable and Realistic Wireless Experimentation Through Physical Emulation," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 63–68, 2004.
[4] M. Cummings and S. Haruyama, "FPGA in the Software Radio," *IEEE Communications Magazine*, vol. 37, no. 2, pp. 108–112, 1999.
[5] "WARP: Wireless Open Access Research Platform," http://warp.rice.edu/trac/.
[6] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker, "Sora: High Performance Software Radio Using General Purpose Multi-core Processors," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2009, pp. 75–90.
[7] S. Papanastasiou, J. Mittag, E. G. Ström, and H. Hartenstein, "Bridging the gap between physical layer emulation and network simulation," in *Proceedings of the IEEE Wireless Communication and Networking Conference*, Sydney, Australia, April 2010.
[8] K. Bilstrup, E. Uhlemann, E. G. Ström, and U. Bilstrup, "On the ability of the 802.11p MAC method and STDMA to support Real-Time Vehicle-to-Vehicle communication," *EURASIP Journal on Wireless Communications and Networking*, no. 2009, December 2008.
[9] Y. Jian and S. Chen, "Can CSMA/CA networks be made fair?" in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. San Francisco, California, USA: ACM, 2008, pp. 235–246.
[10] M. Torrent-Moreno, P. Santi, and H. Hartenstein, "Fair sharing of bandwidth in VANETs," in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*. Cologne, Germany: ACM, 2005, pp. 49–58.
[11] L. Hogie, P. Bouvry, M. Seredynski, and F. Guinand, "A Bandwidth-Efficient Broadcasting Protocol for Mobile Multi-hop Ad hoc Networks," in *Proc. of the Int'l Conference on Systems, Mobile Communications and Learning Technologies*, April 2006, pp. 71–71.
[12] M. Mariyasagayam, T. Osafune, and M. Lenardi, "Enhanced Multi-Hop vehicular broadcast (MHVB) for active safety applications," in *Telecommunications, 2007. ITST '07. 7th International Conference on ITS*, 2007, pp. 1–6.
[13] "Network Simulator ns-2," http://www.isi.edu/nsnam/ns/.
[14] Q. Chen, D. Jiang, V. Taliwal, and L. Delgrossi, "IEEE 802.11 based Vehicular Communication Simulation Design for NS-2," in *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*. Los Angeles, CA, USA: ACM, 2006, pp. 50–56.
[15] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 Modeling and Simulation in ns-2," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*. Chania, Crete Island, Greece: ACM, 2007, pp. 159–168.
[16] A. Kuntz, F. Schmidt-Eisenlohr, O. Graute, H. Hartenstein, and M. Zitterbart, "Introducing Probabilistic Radio Propagation Models in OMNeT++ Mobility Framework and Cross Validation Check with NS-2," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*. ICST, 2008, pp. 1–7.
[17] "The NS-3 Network Simulator," http://www.nsnam.org/.
[18] J. Ryu, J. Lee, S. Lee, and T. Kwon, "Revamping the IEEE 802.11a PHY simulation models," in *Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. Vancouver, British Columbia, Canada: ACM, 2008, pp. 28–36.
[19] "SWANS — Scalable Wireless Ad hoc Network Simulator," http://jist.ece.cornell.edu/.
[20] "GloMoSim — Global Mobile Information Systems Simulation Library," http://pcl.cs.ucla.edu/projects/glomosim/.
[21] J. G. Proakis, *Digital Communications*, 4th ed., ser. in Electrical and Computer Engineering. McGraw-Hill, 2001.
[22] T. Hwang, C. Yang, G. Wu, S. Li, and G. Ye Li, "Ofdm and its wireless applications: A survey," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 4, pp. 1673 –1694, May 2009.
[23] A. F. Molisch, F. Tufvesson, J. Karedal, and C. F. Mecklenbräuker, "A Survey On Vehicle-to-Vehicle Propagation Channels," *IEEE Wireless Communications*, no. 2009, December 2009.
[24] IEEE Standards Association, "IEEE 802.11 LAN/MAN Wireless LANS," http://standards.ieee.org/getieee802/802.11.html, 2007.
[25] International Organisation for Standardisation, "Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model (ISO/IEC 7498-1:1994)," http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html, 1994.
[26] "Simulink simulation and model-based design," http://www.mathworks.com/products/simulink/.
[27] J. W. Eaton, *GNU Octave Manual*. Network Theory Limited, 2002.
[28] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless LANs," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. San Francisco, California, USA: ACM, 2008, pp. 339–350. [Online]. Available: http://portal.acm.org/citation.cfm?id=1409983
[29] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang, M. Chen, and G. M. Voelker, "SAM: enabling practical spatial multiple access in wireless LAN," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. Beijing, China: ACM, 2009, pp. 49–60. [Online]. Available: http://portal.acm.org/citation.cfm?id=1614320.1614327
[30] K. Sundaresan, R. Sivakumar, M. Ingram, and T.-Y. Chang, "Medium Access Control in Ad-hoc Networks with MIMO Links: Optimization Considerations and Algorithms," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 350–365, 2004.
[31] Y. Wu, "Network coding for wireless networks," Tech. Rep., 2007. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=70463
[32] G. Acosta-Marum and M. Ingram, "Six time- and Frequency-Selective empirical channel models for vehicular wireless LANs," *Vehicular Technology Magazine, IEEE*, vol. 2, no. 4, pp. 4–11, 2007.
[33] I. Ivan, P. Besnier, M. Crussiere, X. Bunlon, L. L. Danvic, and M. Drissi, "Influence of channel modeling on V2X physical layer performance," in *Antennas and Propagation, 2009. EuCAP 2009. 4th European Conference on*, Barcelona, Spain, 2010, pp. P03–5. [Online]. Available: http://galayaa.com/EUCAP/data/1848906.pdf
[34] C.-H. Liu, "Design and Evaluation of Energy Detection Algorithms for IEEE 802.11a Systems," in *Radio and Wireless Conference*, August 2003, pp. 63–66.
[35] E. Sourour, H. El-Ghoroury, and D. McNeill, "Frequency Offset Estimation and Correction in the IEEE 802.11a WLAN," in *Vehicular Technology Conference*, vol. 7, September 2004, pp. 4923–4927.
[36] R. Punnoose, P. Nikitin, and D. Stancil, "Efficient simulation of ricean fading within a packet simulator," in *Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd*, vol. 2, 2000, pp. 764–767 vol.2.
[37] P. Alexander, D. Haley, and A. Grant, "Outdoor mobile broadband access with 802.11," *Communications Magazine, IEEE*, vol. 45, no. 11, pp. 108–114, 2007.
[38] "OpenCL - The Open Standard for Parallel Programming of Heterogeneous Systems," http://www.khronos.org/opencl/.
[39] "PhySimWiFi for NS-3 - An IEEE 802.11 OFDM physical layer simulator within NS-3," http://dsn.tm.kit.edu/ns3-physim.php.



**Jens Mittag** holds a diploma in computer sience from the University of Karlsruhe, Germany, and is currently pursuing his Ph.D. within the Decentralized Systems and Network Services Research Group at the Karlsruhe Institute of Technology (KIT), Germany. His research interests include vehicular mobile networks, accurate simulation environments and the modeling of the wireless lower physical laayers. He participated in the NOW: Network on Wheels project (2004-2008) and contributed to the standardization of ITS in Europe as a member of the ETSI specialist task force on *Configuration and Validation of Channel Congestion Control Methods of ITS*.



**Stylianos Papanastasiou** received his combined B.Sc. degree in Computing Science and Mathematics from the University of Glasgow, UK in 2002. He was subsequently awarded his Ph.D. degree in 2006 from the Department of Computing Science at the University of Glasgow for his work titled *Investigating TCP Performance in Mobile Ad Hoc Networks*. Since November 2008 he has been working as a PostDoc at Chalmers University, Sweden in the area of vehicular communications and network simulations. He has published several peer-reviewed journal and conference papers as well as a number of book chapters on the topic of ad hoc and vehicular communications. His current research interests include networking protocol analysis, and the development of precise simulators for vehicular networks.

**Hannes Hartenstein** holds a diploma in mathematics and a doctoral degree in computer science, both from Albert-Ludwigs-Universität, Freiburg, Germany. He is a full professor for decentralized systems and network services at the Karlsruhe Institute of Technology (KIT), Germany, and executive director of the KIT Steinbuch Centre for Computing. His research interests include mobile networks, virtual networks and IT management. Prior to joining the University of Karlsruhe, he was a senior research staff member with NEC Europe. He was involved in the FleetNet - Internet on the Road (2000-2003) and NOW: Network on Wheels (2004-2008) projects, partly funded by the German Ministry of Education and Research (BMBF), he actively participated in the EU FP7 project PRE-DRIVE-C2X (2008-2010), and is now contributing to the follow-on project called DRIVE-C2X. He has been TPC co-chair and general chair of various highly selective ACM and IEEE international workshops and symposia on vehicular communications.

**Erik G. Ström** (S'93–M'95–SM'01) received the M.S. degree from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 1990, and the Ph.D. degree from the University of Florida, Gainesville, in 1994, both in electrical engineering. He accepted a postdoctoral position at the Department of Signals, Sensors, and Systems at KTH in 1995. In February 1996, he was appointed Assistant Professor at KTH, and in June 1996 he joined Chalmers University of Technology, Göteborg, Sweden, where he is now a Professor in Communication Systems since June 2003. Dr. Ström currently heads the Division for Communications Systems, Information Theory, and Antennas at the Department of Signals and Systems at Chalmers. He received the Chalmers Pedagogical Prize in 1998 and the Chalmers PhD Supervisor of the Year award in 2009. Since 1990, he has acted as a consultant for the Educational Group for Individual Development, Stockholm, Sweden. He is a contributing author and associate editor for Roy Admiralty Publishers FesGas-series, and was a co-guest editor for the *IEEE Journal on Selected Areas in Communications* special issues on Signal Synchronization in Digital Transmission Systems (2001) and on Multiuser Detection for Advanced Communication Systems and Networks (2008). Dr. Ström was a member of the board of the IEEE VT/COM Swedish Chapter 2000–2006. His research interests include signal processing and communication theory in general, and constellation labelings, channel estimation, synchronization, multiple access, medium acccess, multiuser detection, wireless positioning, and vehicular communications in particular.