PhyCoNet-Sim: A Framework for Physically Accurate Simulations of Vehicular Ad-Hoc Networks

Steffen Moser, Ralf Schleicher and Frank Slomka
Institute of Embedded Systems/Real-Time Systems
Faculty of Engineering, Computer Science and Psychology
Ulm University, Albert-Einstein-Allee 11, 89081 Ulm, Germany
Email: steffen.moser@uni-ulm.de, ralf.schleicher@uni-ulm.de, frank.slomka@uni-ulm.de

***Abstract*—For decades, the simulation has been a well-established methodology to study the behavior of wireless telecommunication networks. While network and link layer protocols are simulated by using very detailed models, there is typically still a lack of explicit and deterministic considerations of the physical layer. Phyical layer, antenna and radio channel are regularly approximated with simplistic models based on statistic distribution of bit error rate values. While this approach might deliver a sufficient accuracy to compare the performance of routing protocols or other higher-level applications, it disallows, however, studying cross-layer concepts, multi-user communication or the explicit consideration of deterministic channel models. It also denies the physical layer to be an explicit degree of freedom in the design space. To overcome this problem, we set up a PhyCoNet-Sim, a co-simulation framework which links OMNeT++ to a physical layer simulator based on GNU Radio or Matlab/Simulink.**

Keywords*–*VANET; Simulation; Physical Layer; Matlab; Simulink

I. Introduction

Besides field tests and formal verification approaches, simulations are an important methodology to understand the behavior of distributed systems. Especially in the research area of wireless communications in highly dynamic scenarios like Vehicular Ad-hoc NETworks (VANET), simulations are a very effective method to get an in-depth understanding of a complex network's behavior: Compared to field tests, simulations offer fast and reproducible results in an early design phase without the necessity to develop, setup and deploy the systems. In highly mobile scenarios, also formal verification approaches reach their limits: A mobile network's behavior relies heavily on stochastic processes, e.g., mobility behavior, radio channel conditions, and so on. To allow formal verification methodologies to be applied to these systems, a lot of data would have to be gathered and statistically evaluated in the first place by exploiting measurement campaigns. For this reason, simulations play a well-established and major role in VANET research.

The wireless network simulation frameworks, which are typically used today, come with very accurate and deterministic models of the network and link layer protocols. The reason is very obvious: The protocol behavior can be simulated efficiently by using discrete event-triggered simulation engines. Highly performance-optimized frameworks like OMNeT++ [1], NS-3 [2] or JiST [3] rely on this technique. Specialized vehicular ad-hoc network extensions are available, e.g., Vehicles in Network Simulation (VEINS) [4] or Scalable Wireless Ad-Hoc Network Simulator (SWANS). They allow simulating even large-scale scenarios within a reasonable amount of computation time. The simulated link layer or network protocol code can be similar or even identical to implementations used in real-world systems, so aspects like handshaking, queuing and forwarding packets, sending acknowledgments, finding optimal paths, updating routing tables and evaluating properties like channel load, network capacity and packet delivery rates can be done very realistically.

When looking into the internals of state-of-the-art simulation frameworks, one can clearly identify a major problem: The system's components which reside below the link layer (e.g., the components of the physical layer (PHY), the antenna(s) and the radio channel) are modeled in a very simplistic way: A scalar Bit Error Rate (BER) value is used as the only environmental input data to the protocol simulation when the simulator has to decide about the reception of a frame. Using a look-up table, the BER is derived from the Signal-to-Noise Ratio (SNR) value, which is itself based on a fixed transmission power, the distance of the communicating nodes, a stochastic fading model and a background noise level, for example Additive White Gaussian Noise (AWGN). Abstractly spoken, the BER value used by the link layer simulation represents the whole behavior of physical layer, antennas and radio channel with having only the communicating nodes' spatial distance as an environmental input. Slightly improved simulation frameworks extend this approach by using the Signal-to-Interference-plus-Noise Ratio (SINR) to estimate the BER. In SINR-based models, the signal powers of neighboring network nodes are added to the background noise level.

The reasons for using the above-mentioned simplistic models may be found among the following challenges: Physically accurate channel models and signal-based physical layer models do not fit well into discrete event-triggered simulations: The physical layer introduces signal representations, which have to be considered at least in a time-triggered manner, the physics of antenna and radio channel clearly belong to a continuous-time domain. Considering these models explicitly consumes much more computation time compared to classical simulations. Another difficulty is caused by the fact that PHY implementations are done in hardware, so the network protocol designers do not have access neither to the used algorithms nor to their implementations. The simplifications, which are applied, often reach through the whole simulation frameworks: Taking an arbitrary IEEE 802.11p simulation in OMNeT++ as an example, a situation is implicitly considered as a collision at the potentially receiving nodes if more than one node gets into transmission mode in the same time interval within a certain spatial range. While this behavior is the correct one for protocols using a Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) strategy on the Medium Access Control (MAC) layer, it renders many other approaches impossible [5].

The remainder of this paper is structured as follows: In Section II, the limitations of today's ad-hoc network simu-

 ISBN: 978-1-61208-643-9

lators are illustrated. In Section III, we introduce literature where possibilities are discussed to overcome these limitations. Section IV is used to present our approach of co-simulating radio channel, PHY layer and network. In Section V, we evaluate our approach. We close this paper with Section VI, where we summarize and give a perspective to future work. In Section VII, we offer a download link to the framework.

II. Problem Statement

The following list gives an excerpt of four aspects which are hard to explore with state-of-the-art network simulators:

- The state-of-the-art Dedicated Short-Range Communication (DSRC) protocol in vehicular ad-hoc networks is based on IEEE 802.11p. Technically, IEEE 802.11p is an adaption of IEEE 802.11a to vehicular networks, while the IEEE 802.11a standard has been published in 1999. Both, Wireless Local Area Networks (WLAN) and cellular networks have evolved a lot since then, the main reason is that higher-integrated circuits allow much more advanced signal processing schemes. This means, Multiple-Input, Multiple-Output (MIMO) transceivers, Multi-User MIMO (MU-MIMO) transceivers and directed beam forming are common in today's IEEE 802.11ac, which is ubiquitous in homes and offices. The upcoming IEEE 802.11ax standard will contain even more interesting advances in the PHY layer, so there is a distinct interest to evaluate the suitability of these advances for VANETs, especially since limitations of IEEE 802.11p are becoming more and more evident [6]. This is a difficult task with classic network simulations. It would require to gather a lot of statistics data in advance, but even this makes it hard to simulate the behavior realistically because the channel's influence (multi-path propagation, shadowing by buildings) is much higher than in 802.11a or 802.11p. Therefore, it is more difficult to accomplish everything with a single statistics-based distribution. When considering beam forming, there is a cross-layer scenario between PHY layer and network layer [7], [8].
- It is an open question where to place antennas on vehicles. Especially MIMO systems, which necessarily introduce multiple antennas per node, require the number, the form and the position of antennas to be explicit properties of the design space. To determine, for example, which configuration is optimal in specific environmental scenarios (e.g., free-space intersection, high buildings, partly shadowed antennas by trucks), it is necessary that the electrical behavior of antennas to be modeled.
- In ad-hoc network simulations, the SNR or SINR values are calculated by a model of the radio channel physics. The radio channel models used in state-of-the-art simulations are typically rather simple combinations of stochastic fading models on the one hand and distance-based path-loss models on the other hand. The latter parts determine the SNR by an implementation of Friis transmission equation [9] which has been published in 1946. Besides the Euclidean distance of the two communication nodes, Friis' formula considers the gains of the antennas and their effective size based on the wavelength. It does not consider environmental effects like buildings, vehicles and plants. There are, of course, also more sophisticated models available. It has been shown that in highly dynamic network topologies, like VANETs, simple distance-based path-loss models do not offer the required accuracy. This is especially true for urban scenarios where buildings introduce unequally distributed shadowing effects. This has been extensively studied in [10], [11], and [12]. For this reason, ray-optical channel models have been developed which calculate a delay spread of a signal based on the three-dimensional environment. For example, in vehicular ad-hoc networks buildings, vehicles and the terrain roughness are considered by ray-optical approaches. The resulting delay spread shows the temporal diversification and power distribution of different multi-path components. To link ray-optical channel models to traditional wireless network simulators, the delay spread needs to be reduced to a scalar SNR value, effectively discarding most of its information.
- Cellular networks of the fifth generation (5G) are ready to be deployed. The cellular network service providers promise remarkably lower latencies and higher data rates compared to 4G. Due to the fact that cellular network-based applications in vehicles' comfort and emergency systems already exist, the influence of cellular network-based services will presumable increase further. To evaluate a combined usage of cellular and DSRC communication or study protocol convergence concepts in VANET scenarios, a simple IEEE 802.11p-tailored simulation framework is not sufficient anymore.

We present the framework PhyCoNet-Sim, which uses co-simulation of radio channel, physical layer and the upper layers of the distributed system in order to tackle down the restrictions, which lead to the above-described problems. All components are already existing as open-source or commercial tools. For this reason, we don't present a more advanced simulation or approach for a given subcomponent in this paper. Instead, we combine the best approaches available to a co-simulation by designing and implementing suitable interfaces.

III. Related Work

In this section, we present work which is related to the approach presented in this paper. Note that our approach to make the PHY layer an explicit variable in the design phase of a distributed system is intentionally suitable for wireless communication systems in general. However, VANETs are among these kinds of applications, which make the most challenging demands due to the low channel coherence times caused by the high mobility of the transmitting and receiving but also of the neighboring nodes. For this reason, we compare our approach especially to related work in VANET research.

The necessity of enhancing both precision and capabilities of wireless ad-hoc network simulations is not new. Our challenge papers date back to 2010 [13]. The ideas presented were driven by problems which occurred when trying to integrate results of deterministic channel models into VANET simulations. Deterministic channel models typically generate an impulse response, i.e., a distribution of signal power over time which shows the delay-spread of an Dirac impulse and thus the channel's properties. VANET simulators did not offer any interface for an impulse response, as the statistics-based models work with the scalar BER value, only.

In the meantime, different concepts have been proposed. In [14], Papanastasiou et al. present a method for integrating the simulation of the physical layer into network simulations. Their approach is based on the discrete event simulator NS-3 and uses the IT++ library for the transceiver implementation. This approach does not fully conform to the IEEE 802.11p standard since it solely implements the OFDM transmission method, the PHY frame format, the modulation and the coding schemes. As channel model either the path loss or the Rayleigh fading model were employed, and only a rudimentary traffic simulation method was used which did only encode relative

 ISBN: 978-1-61208-643-9

movement and did not take shadowing and other environmental effects into account. This simulator is named PhySim-Wifi for NS-3, is available in the version 1.2, and accessible on the website of the research group. Apparently, there are no more releases since April 2012 and we, therefore, consider the project to be discontinued.

In [15], Judd and Steenkiste describe a Hardware-In-the-Loop-based (HIL) channel emulator. Using a wired connection between the antenna ports of real wireless hardware and a channel emulator based on a Field-Programmable Gate Array (FPGA), they can emulate both, the channels effects on the radio propagation and real PHY behavior. The system supports movement of the simulated nodes by altering the channel conditions accordingly. Interfering signals are superposed. They use a simple path-loss model as a channel model, but the authors point out that simulation of the 3D environment is possible by implementing, for example, a ray-tracing based model on the FPGA. This is a very interesting and flexible approach for accurate physical layer consideration in wireless network simulations: Real network interface hardware is used for signal processing. At the same time this is a major downsides of this concept: It obviously depends on hardware implementations of the protocol, which is about to be studied. Wireless network interfaces must be already available which renders the approach unsuitable in a very early design phase.

Making statistics-based BER calculation more accurate is also still a topic in communication technology research. In [16], Schneider et al. did a large measurement campaign, but used fixed base stations as known from cellular networks. A more detailed campaign, which addresses especially the channel behavior in VANET, was done by Walter in 2016 [17]. He showed that the VANET scenario has a strong impact on choosing the correct statistics distribution.

In [18], Bloessl et al. describe their design and implementation of a Software-Defined Radio (SDR) of IEEE 802.11p using the open-source SDR framework GNU Radio. It has been verified by comparison against commercially available IEEE 802.11p hardware. It contains a complete PHY layer especially for DSRC applications.

Based on Bloessl's work, we developed *Signal Simulation* in 2015, which represents an interface between VEINS and OMNeT++ on the one hand and GNU Radio on the other hand [19]. It was shown that firstly, it works successfully and secondly, for IEEE 802.11p the simplifications present in VANET simulators are valid – at least, when neglecting channel effects and explicit PHY simulation. It is of special interest that the approach is generic and not necessarily bound to IEEE 802.11p, which is only used for proof of concept. This approach solved our requirements from a theoretical point of view completely. Unfortunately, the GNU Radio framework lacks models for the most-recent highly-sophisticated WLAN standards like IEEE 802.11n/ac/ax or for the most recent cellular network standards. When looking at PHY models besides the IEEE WLAN standards, for example, 4G/LTE, one finds models which have been partly developed.

To sum up, using an SDR for co-simulation is the way to go. Having a pure open-source GNU Radio implementation is a promising application to gain deterministic models for the generation and decoding of signals, but has a major drawback regarding the available building blocks.

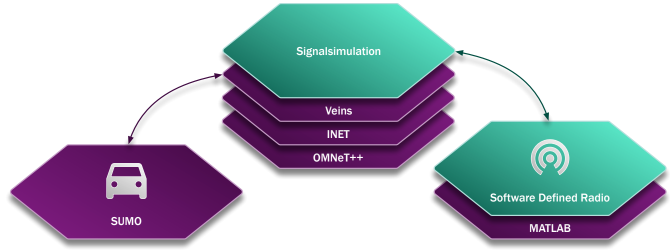

Figure 1. Schema of tools used in PhyCoNet-Sim

IV. PhyCoNet-Sim: Channel, Physical Layer and Network Co-Simulation using a Software-Defined Radio Approach

In this section, we present our approach PhyCoNet-Sim – a co-simulation of channel, physical layer and the network. Essentially, it is an extension of Signal Simulation to allow a flexible substitution of the subcomponents. Figure 1 gives an overview of the components used by PhyCoNet-Sim: VEINS and OMNeT++ are well-accepted ad-hoc network simulators. Simulation of Urban Mobility (SUMO) [20] is a widely-used microscopic vehicular traffic simulator. Matlab/Simulink [21] by The Mathworks is an extremely popular platform in different kinds of engineering disciplines. It comes with a lot of toolboxes. The Communication System Toolbox [22], the WLAN System Toolbox [23] and the Antenna Toolbox [24] exactly address all aspects of the physical layer and the antenna. Systems generated by the help of these toolboxes are specified in a very detailed way, because with the help of additionally available code generators, the toolboxes are actually used in designing and deploying SDRs. For our purpose, we can directly exploit the SDR code to simulate the physical layer. In contrast to GNU Radio, the above-mentioned Matlab/Simulink toolboxes come with implementations of a lot of highly-configurable IEEE WLAN standards, even components of upcoming standards are supported. Regarding radio channel modeling, the Communication System Toolbox comes with a lot of statistics-based methods. As there is a real signal representation for all transmitted packets in the simulation, there is a common interface for integration of impulse responses, which can stem from measurements, statistics-based models and ray-tracing-based models. For the latter, we implemented parts of [10], [12], [25] and [26] to allow deterministic channel behavior for a given environmental input.

In the next sections, the most important components of our PhyCoNet-Sim approach are described. We especially stress the differences between Signal Simulation and PhyCoNet-Sim.

A. Transmission

The MAC layer and all layers above the MAC layer are simulated as in a classical VEINS simulation. We assume that a packet is generated by one of the upper layers, e.g., a Cooperative Awareness Message (CAM) is to be sent. In this case, an according event is generated and queued. The simulation engine jumps to the point in time where the event will take place. As we use CSMA/CA in our IEEE 802.11p reference implementation, it will be checked whether the ready-to-transmit node detects a free channel. If so, the transmission process takes place. Note that the approach works also for non-CSMA MAC protocols, in case of Code-Division Multiple Access (CDMA) or MU-MIMO approaches, other possibilities

 ISBN: 978-1-61208-643-9

to detect when a transmission can be scheduled are possible. Whenever a transmission event takes place, for all receivers within a configurable maximum distance, a reception event is triggered. In a classical VEINS simulation, the receivers within a certain range would get a copy of the transmitted data depending on the BER based on the SNR value. The latter would be depending on the distance of the nodes to estimate the long-term path loss and a stochastic process to estimate the short-term fading. This does not happen in PhyCoNet-Sim.

B. Reception

1) Signal Generation: PhyCoNet-Sim does not deliver the data to the receiving nodes. Instead, in a first step, the position vectors and velocity vectors of the transmitting and the receiving vehicles are stored. This information is necessary as input data for the channel model. A detailed schema of the reception process is depicted in Figure 2. After storing speed and position of the vehicles, the signal, which corresponds to the transmitted frame, is generated. Therefore, the transmitter part of the Matlab-based SDR modem is called via the interface developed for this work. It maps the transmitter's data frame to a signal constellation which is a discrete-time set of baseband signal samples. In an SDR setup these signal samples would be fed to a digital-to-analog converter and later to an up-sampler, a transmission amplifier and to the antenna. In our case, we store the baseband time-domain signal samples in a matrix.

2) Antenna Characteristics and Radio Channel Influence: In order to simulate the system as accurately as possible, the simulated signal must suffer from the antenna's and the radio channel's influence. This is put in execution by convolving the generated transmitter signal with the channel's impulse response. The latter can be generated by a deterministic ray-optical channel model, for example. Using the impulse response defines a very versatile interface which allows different kinds of channel and antenna models to be applied. Instead of using the ray-optical model, it is also possible to use statistics-based channel models available in Matlab's Communications System Toolbox as well as data sets gathered by measurement campaigns [16], [17]. As the velocity and location profiles of the transmitting node and the receiving node are known, it is possible to map the channel effects caused by the movement directly to a non-stationary channel response. This means: Our model allows to consider a node's movement during the transmission process which needs, of course, a non-zero amount of time to transmit a frame of a certain length. The granularity, which is available therefore, is defined by the mobility model used in the simulation. In our test-case, we used vehicular traffic models available in SUMO because there is already a very good integration in VEINS. All signal-processing parts are done in Matlab/Simulink by using the introduced toolboxes. For this reason, the user has access to a large number of building blocks and implemented wireless communication standards. It is also possible to model and simulate complex multi-antenna scenarios using the Antenna Toolbox and consider their behavior.

3) Signal Buffering: After convolving the generated signal with the channel's impulse response, the resulting signal pattern contains all multi-path, delay, Doppler-spread, antenna-gain and AWGN noise information. In a two-node scenario or when using a Time-Division Multiple Access (TDMA) MAC protocol, the signal pattern could be directly forwarded to the Matlab SDR's receiver routine. In a multi-hop scenario with all nodes being unsynchronized this is not possible. Although this process can be very memory-consuming, it is necessary to store the resulting signal pattern in a signal buffer. The reason is that we have to consider other neighboring nodes, which might be transmitting in a time interval which overlaps with the time interval of the first signal we received. Even with a collision avoiding strategy, this can happen due to the high mobility of the nodes. If we simulated only CSMA-based protocols, we could consider such effects as collisions and discard the received signal. In our scenario, we are going to feed the contents of the signal buffer to a Matlab SDR's receiver engine, which allows us to do an unbiased decoding of the signal. Before triggering the receiver engine, other incoming signals have to be treated iteratively as explained here. All overlapping parts are stored in the signal buffer. Compared to our GNU Radio-based implementation, the data structure used for the signal buffer in this work is adaptive. Instead of providing a fixed amount of memory, which can be costly when it comes to large scenarios, the memory is now allocated dynamically.

4) Interferences: Interference is a local phenomenon of wave physics. This means that the superposition of interfering waves has to be calculated for each receiving node. To do so, we use the signal components, which are stored in a node's signal buffer. In our SDR approach, we do not have a mathematical function of a continuous-time signal in our buffer which we could simply add-together. While the channel and antenna effects on the signal could be simulated in a fully continuous-time domain (e.g., by using a computer algebra system), the signal, which leaves our SDR transmitter, consists of discrete-time samples which is the way to go for real systems deployments. For this reason, also the signal in our buffer at the receiver's site is described by samples instead of a closed mathematical function. To avoid any loss of generality, an accurate PHY layer simulation requires that a receiver model is able to cope with signal components which can arrive at arbitrary points in time. Signals, which reach a receiver, can originate from different senders being active at overlapping time intervals or from multi-path propagation. Caused by the finite signal propagation speed of the signals, both cases introduce an arbitrary delay, which is independent from the simulator event steps or any time steps. The SDR models in Matlab do not solve the calculation of the superposition directly. For this reason, interpolation approaches have been implemented as we already proposed in [19]. While linear time interpolation would be very easy to apply to all signal samples in the signal buffer, it introduces an error. In our tests with the Matlab SDR, the rate of reception with respect to a given signal-to-noise ratio was worse when linear interpolation was used compared to using the nearest sample, in which case there was no interpolation. This confirms our previous results. A better solution is an ideal band-limited interpolation by using a fractional-delay filter [27].

5) Decoding: The superposition calculated by the fractional-delay filter is forwarded to the receiver part of the Matlab SDR. It tries to decode the superposition of the different signal components, channel-introduced distortions and the background noise. Depending on the channel coding parameters, a specific amount and distribution of bit errors may be corrected in the PHY layer. After a successful decode, the node's MAC layer will be notified about an incoming data frame. For more advanced decoders (e.g., MU-MIMO in the up-link as proposed for IEEE 802.11ax), there can be several decoding processes, which can take place at the same time.

C. Software-Defined Radio

The SDR modem used for our implementation has been configured by using the WLAN System Toolbox of Matlab

 ISBN: 978-1-61208-643-9

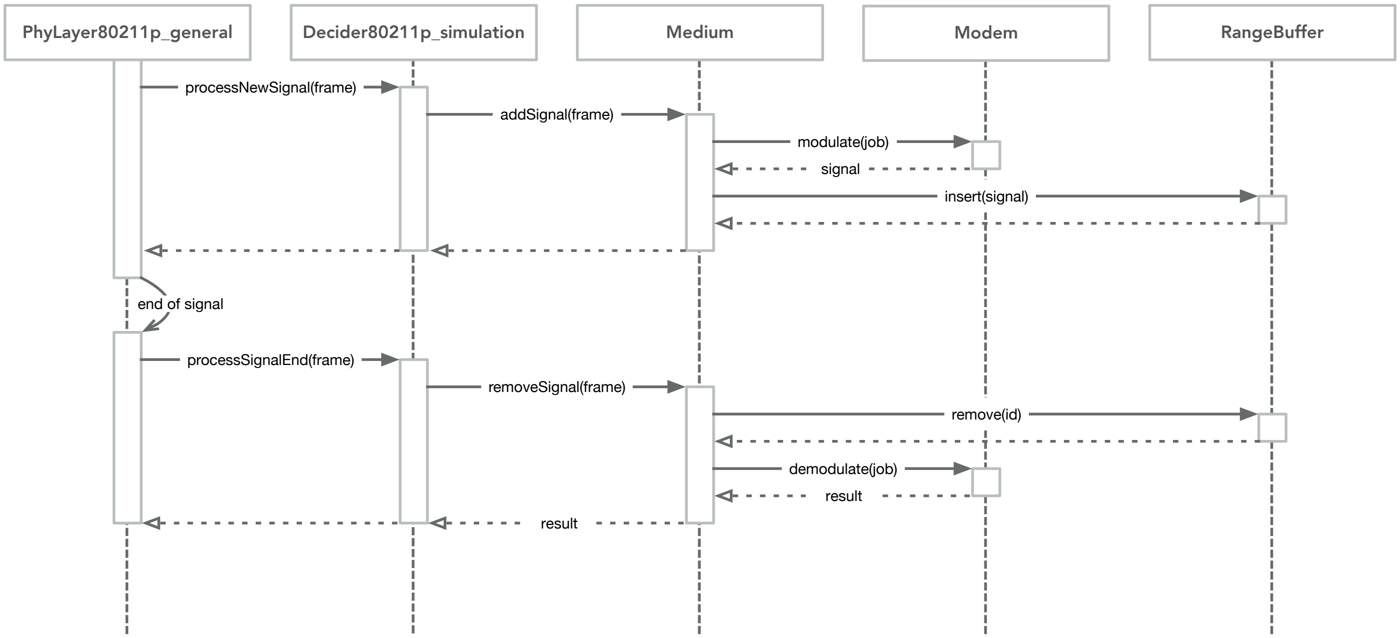

Figure 2. Detailed schema of a signal reception process in PhyCoNet-Sim

2017a. Essentially, the WLAN System Toolbox generates and configures Matlab code for transmitter and receiver. The following supported modem types are supported:

- Type:
  - IEEE 802.11a/b/g/p (Non-High Throughput)
  - IEEE 802.11n (High Throughput)
  - IEEE 802.11ac/ax (Very-High Throughput)
- Channel bandwidths: 10, 20, 40, 80 and 160 MHz
- Modulation schemes: Direct-Sequence Spread Spectrum (DSSS) and Orthogonal Frequency Division Mutliplexing (OFDM)
- Coding schemes:
  - BPSK 1/2, BPSK 3/4, ...
  - QPSK 1/2, QPSK 3/4, ...
  - QAM-16 1/2, QAM-16 3/4, ...
  - QAM-64 2/3, QAM-64 3/4, ...
  - ...

The parameters can be mixed, which means different coding schemes can be combined with various channel bandwidths. The code generated by the WLAN System Toolbox is Matlab code. Thus, it is easily possible to extend it by user-code to evaluate new approaches, which go beyond the implemented standards. Other wireless systems besides WLAN can be designed by using the building blocks from Communications System Toolbox. As a comparison, GNU Radio essentially covers the non-high throughput modems. The execution of the transmitter and receiver functions within the Matlab interpreter offers interesting possibilities for probing signals, but is inefficient. For this reason, we use the Matlab Coder to generate native C++ code when we link it to a network simulation. Essentially, the Matlab coder generates two C++ functions: One for the transmitter and one for the receiver. These functions are linked via an interface to our signal buffer layer in OMNeT++.

V. Evaluation

In this section, we describe the first evaluation steps of the implemented PhyCoNet-Sim. In a first step, we started with an implementation of our GNU Radio-based approach of [19]. We compared it with PhyCoNet-Sim regarding accuracy and performance.

A. Accuracy

To get a an impression regarding the accuracy, we took the IEEE 802.11p and compared the GNU Radio receiver with the Matlab receiver by simulating the probabilities of a successful signal frame reception for different signal-to-noise ratios. The plot is depicted in Figure 3 and shows a very similar trend. The Matlab SDR shows a higher reception quality, i.e., it reaches better decoding probabilities for SNR values between 23 dB and 35 dB. We are currently in the process of investigating the differences.

B. Performance

Highly-accurate physical layer simulations consume a lot of computation time compared to traditional network simulations. Essentially, the major part of the costs of computation depend on the number of signal transmissions and receptions. Assuming that one frame is sent which is received by n neighboring nodes, there is one transmission and n receptions. Another cost driver are the length of a frame and the amount of the delay-spread caused by the channel: A large temporal delay-spread of a signal in the signal buffer causes more discrete-time signal samples which must be considered by the decoder engine. In the GNU Radio approach, a signal consisting of 64 OFDM symbols lead to a computation of 115 ms for a whole transmit-receive-cycle (1 transmit, 1 receive). GNU Radio uses Python for connecting the flow graph of signal processing modules. In the GNU Radio approach, the Python interpreter has been restarted each time a signal is received. We optimized our Matlab SDR approach regarding performance. On a similar computer architecture like it was used for our

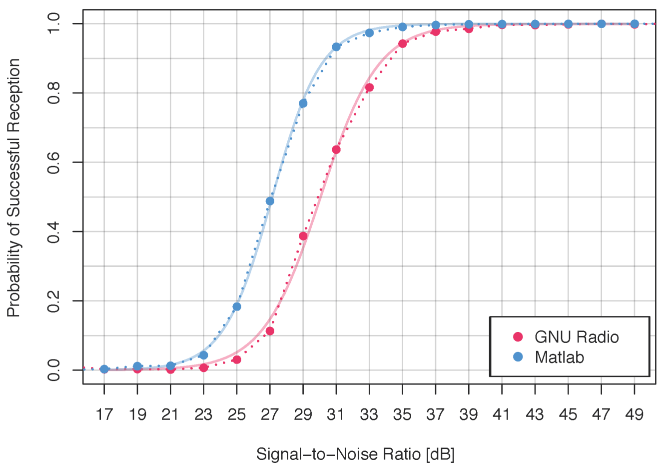

Figure 3. Signal reception probability for the GNU Radio and the Matlab IEEE 802.11p receiver

GNU Radio approach, we got at least an improvement by a factor of 11.

VI. Conclusion and Future Work

The Communication System Toolbox, the WLAN System Toolbox and the Antenna Toolbox offer modern physical layer models as parts of the Matlab/Simulink tool suite. While their primary focus is the development of software-defined radios, we have shown in this paper that the models can be applied very suitably to complex vehicular ad-hoc network simulation scenarios. Therefore, we enhanced an existing approach, which already linked OMNeT++ with GNU Radio, by the ability to delegate the PHY layer calculation to the Matlab toolboxes. We have implemented a prototype for the simulation of IEEE 802.11a/g/p and provided an integration in the VEINS simulator. However, our approach is both, protocol agnostic and application agnostic, i.e., all parts of the co-simulation can be exchanged. In future work, we will use the developed framework for performance simulations for more modern PHY layer mechanisms in vehicular ad-hoc networks. Another interesting research question will be radio convergence (5G and WLAN) in complex applications.

VII. Availability

The PhyCoNet-Sim framework has been integrated into the *WAVE – Next Generation* project. It is available at http://wave-ng.net.

References

[1] "OMNeT++," retrieved: 2018-04-30. [Online]. Available: https://omnetpp.org/

[2] "The Network Simulator NS-3," retrieved: 2018-04-30. [Online]. Available: http://www.nsnam.org/

[3] R. Barr, "An Efficient, Unifying Approach to Simulation Using Virtual Machines," Ph.D. dissertation, Cornell University, 2004.

[4] "Vehicles in Network Simulation (VEINS)," retrieved: 2018-04-30. [Online]. Available: http://veins.car2x.org/

[5] B. Bloessl, F. Klingler, F. Missbrenner, and C. Sommer, "A Systematic Study on the Impact of Noise and OFDM Interference on IEEE 802.11p," in 9th IEEE Vehicular Networking Conference (VNC 2017). Torino, Italy: IEEE, November 2017, pp. 287–290.

[6] B. Bloessl, M. Gerla, and F. Dressler, "IEEE 802.11p in Fast Fading Scenarios: From Traces to Comparative Studies of Receive Algorithms," in 22nd ACM International Conference on Mobile Computing and Networking (MobiCom 2016), 1st ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services (CarSys 2016). New York, NY: ACM, October 2016, pp. 1–5.

[7] H. Stuebing, A. Jaeger, N. Wagner, and S. A. Huss, "Integrating Secure Beamforming into Car-to-X Architectures," SAE International Journal of Passenger Cars- Electronic and Electrical Systems, vol. 4., Jun. 2011, pp. 88–96.

[8] S. Moser, S. Eckert, and F. Slomka, "An Approach for the Integration of Smart Antennas in the Design and Simulation of Vehicular Ad-Hoc Networks," in Proceedings of the International Conference on Future Generation Communication Technology (FGCT), London, UK, Dec 2012, pp. 36–41.

[9] H. Friis, "A Note on a Simple Transmission Formula," in Proceedings of the I.R.E. and Waves and Electrons, vol. 41, 1946.

[10] J. Maurer, W. Sörgel, and W. Wiesbeck, "Ray Tracing for Vehicle-to-Vehicle Communication," in In Proceedings of the URSI XXVIIIth General Assembly 2005, New Delhi, India, Oct. 2005.

[11] I. Stepanov, D. Herrscher, and K. Rothermel, "On the Impact of Radio Propagation Models on MANET Simulation Results," in Proceedings of the 7th IFIP International Conference on Mobile and Wireless Communication Networks (MWCN 2005), Marrakech, Morocco, pp. 61–78.

[12] 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, pp. 161–166.

[13] S. Moser and F. Slomka, "Towards more Realistic Simulations of Ad-hoc Networks - Challenges and Opportunities," in Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Juli 2010, pp. 422–427.

[14] S. Papanastasiou, J. Mittag, E. G. Strom, and H. Hartenstein, "Bridging the Gap between Physical Layer Emulation and Network Simulation," in Wireless Communications and Networking Conference (WCNC), 2010 IEEE. IEEE, 2010, pp. 1–6.

[15] G. Judd and P. Steenkiste, "A Software Architecture for Physical Layer Wireless Network Emulation," in Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization. ACM, 2006, pp. 2–9.

[16] C. Schneider, M. Narandžić, M. Kaske, G. Sommerkorn, and R. Thoma, "Large Scale Parameter for the WINNER II Channel Model at 2.53 GHz in Urban Macro Cell," in Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st, May 2010, pp. 1–5.

[17] M. Walter, "Scattering in Non-Stationary Mobile-to-Mobile Communications Channels," Ph.D. dissertation, Ulm University, 2016.

[18] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "Towards an Open Source IEEE 802.11p Stack: A Full SDR-based Transceiver in GNURadio," in 5th IEEE Vehicular Networking Conference (VNC 2013). Boston, MA: IEEE, December 2013, pp. 143–149.

[19] D. Maier, S. Moser, and F. Slomka, "Deterministic Models of the Physical Layer Through Signal Simulation," in Proceedings of the 8th International Conference on Simulation Tools and Techniques, ser. SIMUTools '15, 2015, pp. 175–182.

[20] "SUMO - Simulation of Urban MObility," retrieved: 2018-04-30. [Online]. Available: http://sumo.dlr.de/

[21] The Mathworks, "Matlab," retrieved: 2018-04-30. [Online]. Available: https://mathworks.com/

[22] ——, "Communications System Toolbox," retrieved: 2018-04-30. [Online]. Available: https://mathworks.com/products/communications.html

[23] ——, "WLAN System Toolbox," retrieved: 2018-04-30. [Online]. Available: https://mathworks.com/products/wlan-system.html

[24] ——, "Antenna Toolbox," retrieved: 2018-04-30. [Online]. Available: https://mathworks.com/products/antenna.html

[25] A. Schmitz and L. Kobbelt, "Wave Propagation Using the Photon Path Map," in PE-WASUN '06: Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks. ACM Press, 2006, pp. 158–161.

[26] J. Nuckelt, M. Schack, and T. Kürner, "Deterministic and Stochastic Channel Models Implemented in a Physical Layer Simulator for Car-to-X Communications," Advances in Radio Science, vol. 9, no. C. 5-1, 2011, pp. 165–171.

[27] V. Valimaki and T. I. Laakso, "Principles of fractional delay filters," in 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100), vol. 6, 2000, pp. 3870–3873 vol.6.