A Tool for Raytracing Based Radio Channel Simulation

Thomas Nitsche
Institute for Informatics
Technische Universität München
Boltzmannstr 3, Munich, Germany
nitschet@in.tum.de

Thomas Fuhrmann
Institute for Informatics
Technische Universität München
Boltzmannstr 3, Munich, Germany
fuhrmann@in.tum.de

ABSTRACT

In this paper, we briefly report on our ongoing work to extend the GNU Radio software suite with a ray tracing based radio channel simulator. Radio channel simulation is an important aspect in the design and evaluation of wireless protocols because noise and interference can have a crucial impact on the performance of a protocol. However, the calculation of radio wave propagation is computationally demanding. Thus, most network simulation frameworks rely on simple statistical radio channel models that do not account for site-specific propagation characteristics. So, these simulators miss important details that might impact a protocol significantly in specific propagation environment.

Our radio channel simulator uses ray tracing techniques to overcome this limitation. It precomputes the channel characteristics of a given scene so that it can then efficiently simulate the corresponding links. Our implementation seamlessly interfaces with the GNU Radio software defined radio (SDR) framework, replacing its statistical channel simulation component. Furthermore, using GNU Radio's various modulation components, our simulator can provide a complete PHY layer simulation interface for other simulators such as ns-3 or OMNeT++. All in all, our simulator enables SDR developers and wireless protocol engineers to quickly assess their design in more realistic simulated environments.

Categories and Subject Descriptors

B.4.4 [**Input/Output and Data Communications**]: Performance Analysis and Design Aids—*Simulation*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

General Terms

Design, Experimentation

Keywords

Software Defined Radio, Network Simulation, Wireless Communication, Channel Model, GNU Radio

SIMUTOOLS 2011, March 21-25, Barcelona, Spain

DOI 10.4108/icst.simutools.2011.245555

1. INTRODUCTION

The simulation of wireless links is important for the design and evaluation of wireless network protocols because wireless communication is subject to manifold radio wave propagation phenomena that affect the links, e.g. reflection, scattering, and diffraction. Such phenomena lead to noise and interference effects, which may impair the transmission significantly. Flat fading, for example, can diminish the entire received signal's quality. Frequency selective fading blanks out certain frequencies of the transmitted signal. Under certain circumstances, the said phenomena can also improve the signal quality e.g. through positive interference. Therefore, analyzing the behavior of a protocol in a realistic propagation scenario is of great interest for wireless protocol engineering.

Most network simulators use simple statistical models that do not consider the site-specific geometry; for example, they simply assume that the packet loss ratio increases linearly in an annulus that determines the transmission range. Others use a heavily simplified representation of the environment's geometry to account for the signal propagation effects, for example, the Two-Ray Ground path loss model or the Knife-edge diffraction model. Others do not account for the scenario specific geometry at all for instance the Log-Distance model, the Three-Log-Distance model and the Rician and Rayleigh Fading models. Such models provide an efficient way to simulate the averaged disturbances on wireless links. However, due to the complexity of the propagation phenomena, there are many propagation scenarios where the effects on the transmitted signal combine in a unique way. Therefore, simulations that do not account for site-specific geometries might lack important details. We believe that realistic propagation scenarios can only be simulated adequately with models that explicitly consider the actual propagation paths of the signal.

However, explicit modeling of wireless signal propagation is a complex topic with high computational demands. Different methods from the field of geometrical optics have been proposed in order to explicitly model the propagation effects of wireless signals. None of these methods is implemented into a freely available open source tool that extends the simple wireless models of current network simulation frameworks. Additionally, exact wireless link models require the full simulation of the MAC and PHY layer. However, PHY and MAC layer processing is typically handled by specialized signal processing hardware, which complicates the replacement of the simple statistical wireless link models used in common networking simulations further.

With the emerging trend to software defined radio (SDR) there arise new possibilities for accomplishing this task. SDR attempts to move most parts of the PHY layer signal processing into software. By utilizing source code libraries developed by the software defined radio projects, it is possible to create an almost complete software simulation of the PHY layer. Only the pretty general hardware for the signal mixing and the A/D conversion, which is required before the signal can be fed into the antenna, has to be modeled additionally. For a complete simulation of a wireless link, the signal changes induced by this hardware have to be modeled, too.

Our simulator uses the PHY layer signal processing from the GNU radio project to calculate the wireless signal. Furthermore, it applies the effects caused by the mixing and A/D conversion hardware. Finally, it hands the signal to a ray tracing based signal propagation component, which models multipath propagation and various other physical signal propagation effects. In order to overcome the computational limitations, which generally complicate the calculation of site-specific deterministic signal propagation, we precompute the channel characteristics. This allows us to reduce the number of computationally complex ray tracing processes because the signal transmitted over the wireless channel can be determined efficiently by a convolution of the transmitted signal and the precomputed channel characteristic. Altogether, our simulator explicitly predicts the received signal including many of the site-specific disturbances that appear on the wireless channel.

2. RELATED WORK

Due to the importance of wireless short range communication links, common network simulation frameworks have long extended their models to also incorporate wireless links. These are, however, simple statistical models that ignore the site-specific details of the simulated link. Typically, a simple threshold governs the decision if the reception quality suffices to decode a packet: If the received signal strength is sufficient, the packet is received without any errors, otherwise it is dropped entirely. More sophisticated models reduce the packet reception probability linearly between two thresholds. Furthermore, the signal strength is typically only a function of the sender-receiver distance. Some simulators include an additional angular dependence, but there is no explicit modeling of the signal propagation effects such as reflection and interference in a particular environment.

As, for instance, Bingmann et al. point out [2], ns-3 provides various models of path loss, shadowing effects and fast fading for the simulation of WiFi and WiMax links. Path loss models provided by ns-3 include the Friis free space path loss model, the Log-Distance model, and the Three-Log-Distance model. But all of them only predict the statistical path loss that can be found in typical propagation environments without paying respect to the actual propagation scene. The same principle holds for ns-3's shadowing and fast fading models, which allow to predict packet based link quality measures such as the SNR, the signal to interference and noise ratio (SINR), and the bit error rate (BER). ns-3 only uses empirically determined thresholds for these rates; they determine if a packet is successfully transmitted or has to be dropped. According to Merz et al. [7] ns-3 lacks the modeling of the RF front end, antenna and signal propagation as well as an accurate packet detection and timing acquisition. Merz et al. also propose ray tracing techniques for modeling the signal propagation effects.

Another popular network simulator is OMNeT++. PHY layer and radio channel simulation in OMNeT++ is provided by the Mixim framework [15], which is a consolidation of various wireless link modeling tools formerly used with OMNeT++. Mixim provides a design for an extensible modular PHY layer and radio channel simulator. It allows the combination of different implementations for certain aspects, e.g. radio hardware, analogue models or decider modules. Decider modules calculate the BER of a signal and decide if a received signal is classified as noise or a validly transmitted packet, similar to the decision process in ns-3. Analogue models are in charge when it comes to the simulation of path-loss, shadowing and fading. The Mixim framework provides some implementations for analogue models. A quite sophisticated one is the implementation of the IEEE 802.15.4a channel model as proposed in [8]. Similar to ns-3, Mixim only provides statistical wireless channel models. It does not explicitly simulate the signal propagation in the geometries of the different scenarios.

In order to pay respect to the characteristic propagation effects of a specific propagation scenario, techniques from the field of computational optics are promising. These techniques allow the modeling of fading effects that constitute the propagation effects of a specific scene. Ray tracing techniques have already been used for this purpose since the 1990s. For instance, Ikegami et al. [4] used ray tracing techniques for the prediction of path loss in urban mobile radio systems. More recently, Rautiainen et al. [11] used ray tracing techniques on an full 3D model of a urban scenario. They predict delay and path loss for a carrier frequency of 2.1 GHz and compare the result to real world measurements. Their results show that the characteristic effects of wave propagation are simulated quit accurately, with delay mean errors of 50-100ns and mean path loss errors between 0.1-1dB. However, both presented approaches suffer from their high computational demands. They can therefore not be used in conjunction with network simulation frameworks. Furthermore, they are concerned with medium scale propagation scenarios like urban areas or vehicular ad hoc networks. In these scenarios, the objects considered for ray propagation are buildings or vehicles. This level of detail does not suffice for short range simulations.

With the increasing popularity of short range communication systems, the need for accurate channel simulations has increased. For example, Peter et al. [9] analyze ray tracing methods for the simulation of 60 GHz indoor broadband channels. They study an environment that is similar to ours, but the simulated signal frequencies are significantly higher than in our proposed simulator, which primarily aims at the 2.4 GHz IMS band. Therefore, the effects that Peter et al. model might not all be relevant for our considerations. Wahl et al. [14] predict radio channel characteristics in the 5.9 GHz band in and around vehicles. Their methods face propagation effects that we also expect in our scenario. They compare their results to predictions made with the FEKO tool suite [3], which uses the *Method of Moments* (MoM) to solve Maxwell's equations for electromagnetic waves. Their comparison shows that the predicted results are nearly the same, while the ray tracing approach is more efficient. The ray tracing technique needs about ten times less memory, and simulations calculated in hours using the MoM approach need less than a minute using the ray tracing approach.

However, none of these simulators are integrated into a common network simulation framework probably because network simulations do not only consider pairs of transceivers, but a potentially large number of nodes. Furthermore, models focusing on indoor propagation scenarios, such as those by Peter et al. and Wahl et al. have an even higher computational demand due to their higher complexity level of the simulated scene. Thus, their models allow the simulation of a single indoor channel whereas the calculation of many channels among a group of nodes, as necessary for network simulations, easily exceeds the available computational resources.

Schmitz and Wenig [13] propose a ray tracing enhanced high accuracy wireless link model for ns-2. They precompute the signal propagation characteristics to handle the computational demands, an approach similar our proposed scheme, described in chapter 3 Another viable technique to speed up ray tracing based wireless link models is the use of graphic processors (GPUs), which has, for example, been proposed by Bai and Nicol [1]. Designing our ray-tracing code to use GPUs could speed up the precomputation step of our simulation; however, we have not yet done so because we wanted to keep our simulator design simple and avoid a dependence on specific graphics hardware.

Lewandowski et al. [6] have put forth an approach to extend the wireless link models for the 2.4 GHz band that are used in common network simulation frameworks. They propose to calculate a channel model using a proprietary tool called *Radiowave Propagation Simulator* (RPS). RPS replaces the statistical PHYs-Layer simulation of OMNeT++. Schmitz et al. [12] extend the NS-2 network simulation framework for handling networks of wireless links that range from the size of 802.11 like wireless local area networks up to urban cell networks. They use a ray tracing related approach called *Photon Path Mapping* (PPM) for their simulation.

[13], [6] and [12] seem to be very close to our proposed simulator. However, there is a significant difference. Lewandowski et al. use the wireless channel propagation characteristics as computed by the RPS tool to get a more detailed view on the signal strength received from the different sources in a scene. But they stick to the calculation of the signal to interference ratio as a measure to describe the disturbance of the received signal. The same holds for Schmitz et al. They enhance the NS-2 field strength lookup with data that they calculate with their proposed PPM algorithm. NS-2 later on uses this data for a threshold based decision on the reception or loss of the packet that has been sent over the wireless link. Our simulation, in contrast, models the actual signal that is modified through the various propagation effects. Thus, we can study how a packet might become corrupted on the link; and we can study the details of how two simultaneous transmissions interfere with each other. We believe that these additional details are especially important for PHY layer and cross layer protocol engineering.

3. SIMULATION OF RADIO SIGNAL PROPAGATION

Our simulator consists of two components. The first one precomputes the *power delay profiles* (PDP) for the links between every pair of simulated transceivers. It uses a ray tracing approach to do so. The PDP of a link lists the delay and attenuation for every path the signal can take from the transmitter to the receiver. The second component uses this precomputed information to efficiently calculate the actually received signals for each transmitted frame.

In this section, we describe the first component of our simulator, i.e. the ray tracing approach that we use to calculate the characteristics of every path that the received signal can take, and we explain which signal propagation phenomena we model for predicting these different paths.

3.1 Ray Tracing Approach

As stated in the introduction, we aim at simulating the effects that the obstacles in a given simulation scenario cause. Therefore, we have to consider all the relevant signal propagation phenomena that govern our propagation scenarios e.g. reflection, diffraction, scattering, signal path loss and antenna characteristics. Some of them attenuate the signal; others modify the signal's propagation direction. Interference, in contrast, results from different signals reaching the receiver at the same time. Our signal propagation model incorporates this by considering different rays reaching the receiver on different paths. Such multipath propagation induces the various interference effects, e.g. fading. Also, signals from different sources reaching a receiver at the same time lead to collisions of the transmitted frames. However, this kind of interference does not affect the radio wave propagation component of our simulator. It is handled in the link simulation component.

We use a ray tracing approach to calculate the signal propagation behavior. Our simulator takes a description of the obstacles that are present in a scene as well as the position of all transmitters and receivers. Based on this information, we cast virtual rays from each transmitter position and trace them on their way through the simulation scene. To this end, we decompose each ray into segments. Each segment represents the ray's way between two obstacles. Upon hitting an obstacle, the next segment is calculated in regard to the modeled signal propagation phenomena. Segments hitting the location of a receiver induce a propagation path that later influences the received signal. As the propagation of every virtual emitted ray is uninfluenced by other rays, the different rays can be computed in parallel.

In order to model the wireless link between two transceivers in the second component of our simulator, we need to know all the propagation paths between these two transceivers. We represent this information by the PDP, which contains the delay and attenuation for all the paths that contribute to a link. From the delay, for example, we can then calculate the phase offsets of the paths, which determine the interference of the signal at the receiver.

The delay depends solely on the length of the ray's propagation path:

$$D = c \cdot x \qquad (1)$$

where x is the length of the rays path from sender to receiver and c is the speed of light.

The attenuation for every signal component that reaches the receiver depends on different effects. Most importantly, whenever the signal encounters an object, it loses energy by reflection and diffraction. Furthermore, the signal looses energy through free space path loss.

We model the reflection loss by adding an attenuation factor to every modeled surface. According to Rappaport [10], this factor depends on the incident angle and the obstacle's material properties. The intensity of every reflected ray segment decreases according to this attenuation factor. Diffraction introduces a similar effect. The intensity of the diffracted segment can be calculated by multiplying the incident ray's intensity with a corresponding diffraction attenuation factor.

Correctly modeling the free space path loss is more challenging. According to Rappaport, the received signal power $P_r(d)$ can be calculated as:

$$P_r(d) = K \cdot \frac{P_t \lambda^2}{d^2} \qquad (2)$$

where d is the distance from the source, and K is a constant that contains antenna gains and energy loss in the transmission system; P_t is the transmitted power. One might be surprised that equation (2) contains a factor d^{-2} not d^{-4}. The reason is that with growing distance from the sender, the number of rays that hit a unit volume of the simulation space decreases. This is due to the discrete nature of our ray tracing approach. It corresponds to the fact that the signal power in vacuum reduces with d^{-2}. Thus, implicit by the ray tracing approach less signal power is carried to the simulation space that is further away from the ray's source, and we do not need to model that part of the free space path loss explicitly.

3.2 Ray Propagation Effects

When modeling reflection, we have to differentiate between flat and curved surfaces because flat surfaces just change the propagation direction of the radio signal, but curved surfaces also widen the ray such that the signal becomes diluted. We also surveyed scattering for our simulation. However, we believe that the scattering effects that are relevant on the small scales of our envisaged scenarios are already incorporated sufficiently by our reflection model.

Reflection and diffraction influence the propagation paths of the rays. According to the laws of reflection, each incoming ray is split into two parts. One part is passing through the medium discontinuity at the obstacle surface. The other part is deflected, where the angle of the outgoing ray equals the incidence angle (with respect to the surface normal). The reflection model has to also incorporate the material property of the obstacle surface. The reflection coefficient for a given material characterizes the ratio between the amplitude of reflected ray and the incident ray.

Our simulation only tracks the reflected part of the ray because we assume that the other part gets absorbed by the obstacle. The reflection process is tracked until the ray has lost most of its energy. Tracing the ray further would have negligible influence on the simulation result. In practice, that cut-off leads to a maximum number of reflections for each ray. The higher that number, the better the simulation accuracy and the higher the computational demands.

The simplest form of the laws of reflection only accounts for planar obstacles, i.e. objects with a flat surface. To capture the effect of curved obstacles, we extend each ray into a 5-tuple consisting of the direction $\vec{k}$, the energy intensity I, the beam width α, and the covered distances d and d' of the ray. Here, d represents the length of the entire path that the ray has traveled from its source, whereas d' is the distance that it has covered since the last reflection.

In a 2-dimensional model, the ray direction corresponds to an angle: $\vec{k} \simeq \phi$. Upon reflection at a curved surface, the beam width α increases: $\alpha' = \alpha(1 + \frac{d'}{R})$, where R is the radius of curvature of the surface at the point of incidence. When α exceeds a given threshold, we split the beam. Typically, each reflection at an object with small radius of curvature, e.g. a ventilation pipe, causes a *beam split*.

A beam split attributes equal shares of energy to the resulting rays. Assume, for example, that after a reflection the beam width exceeds the threshold by a factor of three. Then we create three new rays, each obtaining one third of the initial ray's beam width and intensity.

A further propagation effect that may lead to multipath propagation is diffraction. Roughly speaking, it allows the rays to bend around the edges of obstacles. In general, the intensity of diffracted rays is very low. In a scenario that is characterized by many reflections, the intensity contributed through diffraction is negligible. However, in a completely shadowed region the diffracted signal might be the only one received at all. Thus, depending on the occurrence of completely shadowed regions in a propagation scenario, modeling of diffraction might be needed nonetheless. Kouyoumjian et al. [5] describe an elaborated geometrical optic diffraction model that can predict diffracted signals in this case.

Another relevant aspect that has to be considered besides the propagation effects are the antenna characteristics. Real world antennas do not emit the signal energy uniformly. Often, they have a preferred direction to which they emit most of the transmitted signal strength. We model this effect by increasing the emitted number of rays in the according direction.

Figure 1 shows a 2D sample attenuation map for a simulated scenario including an isotropic ray source and three rectangular obstacles. The attenuation is measured in a logarithmic scale. The sender emits an unmodulated carrier frequency at 2.4 GHz. Reflection at the obstacles leads to multipath propagation and thus to carrier interference. While the shown plot tracks multipath propagation for every cell of the simulated scene, further use in the channel simulator only requires the propagation paths between the given transceiver positions.

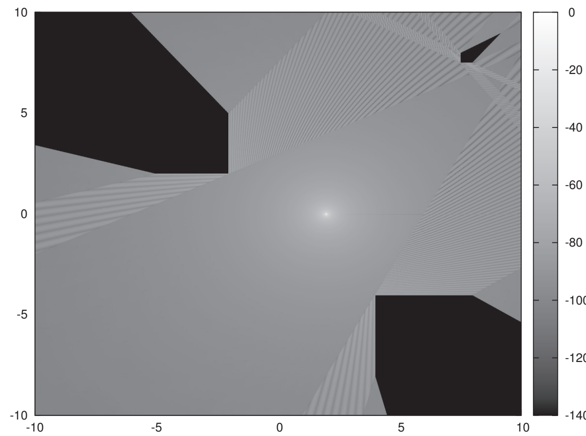

Figure 1: Example of a 2D ray-tracing scene including an isotropic ray source and three rectangular obstacles

4. WIRELESS LINK SIMULATION

The second part of our simulator predicts the signal transmission over a site-specific wireless link. It is designed to interface with the *GNU Radio* SDR framework. There, it replaces GNU Radio's statistical channel simulation component. Making the site-specific channel simulation available as GNU Radio control block allows the easy combination with different GNU Radio signal processing blocks.

To simulate a wireless link, we do not only have to model the wireless signal propagation, as described in section 3. We also have to consider the PHY layer signal processing because it determines the signal that is actually transmitted over the air. In most networking solutions, the PHY layer is implemented in hardware using specialized signal processors. This makes it hard to create an exact PHY layer simulator that precisely determines the RF signal as it is emitted by the different kinds of wireless networking hardware.

The emerging trend to software defined radio simplifies the simulation of the PHY layer because SDR implements most of the signal processing in software. Moreover, open source SDR projects such as GNU Radio enable us to use these software implementations directly for the simulation of the PHY layer. There is only a small part of the transmitter hardware that still has to be modeled to calculate the transmitted signal. It includes the frequency mixing, which transforms the baseband signal (provided by SDR) to the carrier frequency. It also includes the effects of A/D and D/A conversion. The timing and frequency effects of the mixer and the converter highly depend on the quality of the oscillator that is used in the hardware. In of-the-shelf RF hardware, these oscillators may introduce inaccuracies that have a non-negligible impact on the transmitted signal. In order to account for such effects, our simulator models both, the carrier frequency divergence and the sampling frequency divergence.

The said frequency divergence always has to be considered between the transmitter and the receiver hardware. It causes two effects: First, the carrier frequency divergence results in a rotation of the complex baseband signals that the GNU radio modulation code generates. Secondly, the sampling frequency divergence results in a sampling skew between the transmitter and the receiver. This effect leads to the acquisition of samples from the signal at points that lie between the intended sampling positions. In order to model this effect, we resample the signal, i.e. we interpolate the signal samples with the *sinc* function.

Both effects are effects of the frequency divergence between the oscillators of the transmitter and receiver. Thus, one might think that the effect must be treated separately for each transmitter-receiver pair. Luckily, we can split that treatment into two distinct operations: one that adjusts the divergence between the transmitter and a virtual system-wide reference clock and one that adjusts between that clock and the receiver.

The described adjustment is applied to the samples of all transmissions in order to get them on a unique sampling basis for further calculations. Without that adjustment the series of complex samples would continuously oscillate between the constellation points.

After the resampling, the samples constitute the signal as it is transmitted over one path between the sender and the receiver. To account for multipath effects, the signal must be convoluted with the channel impulse response as it is stored in the PDP (cf. section 3). Note that different paths lead to different a rotation of the signal, but they do not require any further resampling.

We save the resampled signals of the transmitter in a data structure called *sample pool*. For every *receiver listening interval* the simulator requests all potentially received transmissions from the sample pool. The receivable transmission can be determined by the transmission start time and the maximal delay that is saved in the PDP for the corresponding transmitter receiver pair.

After having been convoluted, all the receivable transmissions are combined in order to account for the interference between the different senders. These interference effects are frequency dependent, but processing the complete transmitted signals implicitly considers these effects. They can be modeled by simply summing up the complex samples that are received at the same time. Then, the summed up signal is processed according to the receiver's RF-hardware characteristics. This includes the resampling that accounts for the frequency divergence between the receiver and the reference clock. Finally, the samples are handed to the GNU Radio signal processing component, where they are further processed, demodulated and decoded (cf. fig 2).

Besides extending GNU Radio's channel simulator, we also aim at providing a full PHY layer simulation interface for network simulators. Therefore, we extend our simulator with the modulation functionality from the GNU Radio framework.

We realize this by an independent simulator application that reads the MAC layer payload data and configuration options from an XML-style file interface. The application follows the same simulation approach as described above. In addition to that, here, we first need to create the modulated baseband samples from the MAC layer payload data. To do this, we pass the data to external GNU Radio scripts via the GNU Radio file sink mechanism. This enables us to provide different modulations for the PHY layer simulation interface. The supported modulation schemes can be easily extended by creating further GNU Radio scripts. This simulator design also allows us to use other external modulation implementations, as long as they can pass data in the GNU Radio file sink format.

We use the same approach for the demodulation. The samples that constitute the received signal are handed back to external demodulation scripts, which then write the data to an XML output file that can be further processed by a higher level protocol simulator. It is up to the higher level simulator to decide how the modified MAC layer data is handled. Typically, it would verify the frame check sequence, apply an error correction mechanism, etc.

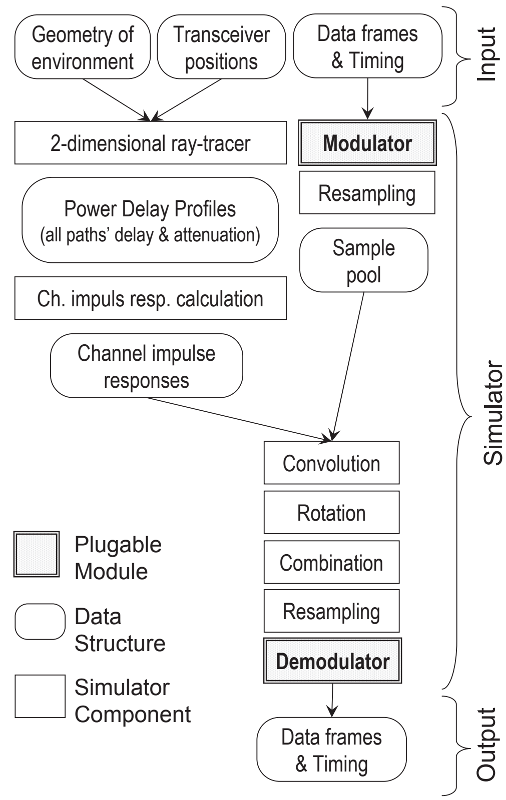

Figure 2: Architecture overview

5. STATE OF THE IMPLEMENTATION

In its current implementation state, our simulator is able to calculate simple 2-dimensional propagation scenarios, while some of the described features still have to be implemented. The basic concept of providing a PHY layer simulation interface that uses XML formatted files containing MAC layer payload data is already operational. Moreover, the simulation configuration options and scene descriptions can also be passed via this XML files. Up till now, the simulator can calculate the received and demodulated payload data for a basic QPSK modulation scheme. The radio wave propagation component supports multiple reflections on flat surfaces, but it still lacks the modeling of diffraction.

Currently, we focus on completing the still missing features, in particular, we work towards a tighter integration with the GNU radio framework. This would ease the composition of modulation schemes that can easily pass their data into our channel modulation component by using GNU Radio's signal processing block architecture.

We are still undecided if we should replace our 2-dimensional propagation model with a 3D model. While we are optimistic, that our 2D component shows the characteristic effects that prevail in multipath propagation scenarios, we need further measurements with actual RF hardware to be able to verify this assumption. If needed, we will extend our simulator with a 3D component or, alternatively, allow the optional import of data provided by other wave propagation simulators e.g. the proprietary RPS. Besides verifying our simulation results on actual hardware, we will also analyze the significance of diffraction effects in wireless short range communications. If theses effects cause significant differences between our simulation results and our real world measurements, we will extend our signal propagation model accordingly.

6. CONCLUSION

In this paper, we have presented our ongoing work on a ray-tracing based network simulator. Unlike other network simulators, we model the effects of radio propagation such as multi-path propagation and frequency selective fading. Thereby, we can study, for example, partial packet corruption and the interference among several transceivers. Moreover, we can study the effects of particular environments such as factory floors.

In contrast to other radio propagation simulators, we focus on the efficient handling of entire networks of transceivers. To this end, we precompute the channel characteristics for all transceiver pairs in a given simulation scenario using a ray-tracing approach. The convolution of this precomputed data with the transmitted signals then yields the received signals, including all interference effects.

Our simulator is closely coupled to the open source SDR framework GNU Radio. Thereby, we can re-use their implementations of the various modulation and channel coding algorithms. Conversely, our simulator can replace GNU Radio's statistical channel model.

Altogether, our work provides wireless protocol engineers with a complete PHY layer and channel simulator, which helps them to quickly evaluate their designs in more realistic simulation environments than current network simulators do.

7. ACKNOWLEDGMENTS

This work has been supported by the German Federal Ministry of Education and Research under grant number 01IS09040.

8. REFERENCES

[1] S. Bai and D. Nicol. Acceleration of wireless channel simulation using GPUs. In *Proceedings of the 16th European Wireless Conference*, pages 841 –848, Lucca, Italy, 2010.

[2] T. Bingmann and J. Mittag. An overview of PHY-layer models in ns-3. `http://www.nsnam.org/workshops/wns3-2009/talks/mittag-workshop.ppt`, March 2009.

[3] EM Software & Systems. Feko - Comprehensive Electromagnetic Solutions. `http://www.feko.info`.

[4] F. Ikegami, T. Takeuchi, and S. Yoshida. Theoretical prediction of mean field strength for urban mobile radio. *IEEE Transactions on Antennas and Propagation*, 39(3):299 –302, March 1991.

[5] R. Kouyoumjian and P. Pathak. A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proceedings of the IEEE*, 62(11):1448 – 1461, 1974.

[6] A. Lewandowski, V. Köster, and C. Wietfeld. A new dynamic co-channel interference model for simulation of heterogeneous wireless networks. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools'09)*, Rome, Italy, 2009.

[7] R. Merz, C. Sengul, and M. Al-Bado. Accurate physical layer modeling for realistic wireless network simulation. `http://www.nsnam.org/workshops/wns3-2009/talks/rmerz_wns3_09.pdf`, March 2009.

[8] A. F. Molisch, K. Balakrishnan, C. Chong, S. Emami, A. Fort, J. Karedal, J. Kunisch, H. Schantz, U. Schuster, and K. Siwiak. IEEE 802.15.4a channel model - final report. Technical report, IEEE 802.1504-0062-02-004a, 2005.

[9] M. Peter, W. Keusgen, and R. Felbecker. Measurement and ray-tracing simulation of the 60 GHz indoor broadband channel: Model accuracy and parameterization. In *Proceedings of the 2nd European Conference on Antennas and Propagation (EUCAP'07)*, Edinburgh, UK, November 2007.

[10] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 2001.

[11] T. Rautiainen, G. Wolfle, and R. Hoppe. Verifying path loss and delay spread predictions of a 3d ray tracing propagation model in urban environment. In *Proceedings of the 56th IEEE Vehicular Technology Conference (VCT 2002-Fall)*, volume 4, pages 2470 – 2474, 2002.

[12] A. Schmitz and L. Kobbelt. Wave propagation using the photon path map. In *Proceedings of the 3rd ACM International Workshop on Performance Evaluation of Wireless Ad hoc, Sensor and Ubiquitous networks (PE-WASUN '06)*, pages 158–161, Terromolinos, Spain, 2006.

[13] A. Schmitz and M. Wenig. The effect of the radio wave propagation model in mobile ad hoc networks. In *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems (MSWiM'06)*, pages 61–67, Terromolinos, Spain, 2006.

[14] R. Wahl, M. Layh, and T. F. Eibert. Wave propagation inside and around vehicles in dynamic time variant scenarios. In *Proceedings of the 63rd IEEE Vehicular Technology Conference (VTC 2006-Spring)*, pages 2883–2886, May 2006.

[15] K. Wessel, M. Swigulski, A. Köpke, and D. Willkomm. Mixim: the physical layer an architecture overview. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools'09)*, Rome, Italy, 2009.