

# Simulation of 802.11 PHY/MAC: The Quest for Accuracy and Efficiency

Michele Segata  
DISI – University of Trento  
Trento, Italy  
Email: msegata@disi.unitn.it

Renato Lo Cigno  
DISI – University of Trento  
Trento, Italy  
Email: locigno@disi.unitn.it

**Abstract**—The goal of this work is to highlight and explain the limitations of traditional physical channel models used in network simulators for wireless LANs, with particular reference to VANETs, where these limitations may jeopardize the validity of results, specially for safety applications. The fundamental trade-off is between simulation time and realism. Indeed, a simulator should provide realistic results as fast as possible, even if several nodes (i.e., hundreds) are considered. Our final goal, beyond this initial contribution, is the development of a stochastic channel model which improves reliability of simulations while increasing computational complexity only marginally. The design of our model is based on the representation of the packet decoding procedure as a Markov Decision (Stochastic) Process (MDP), thus avoiding the computational complexity of the simulation of the entire transmission – propagation – decoding chain bit-by-bit, which can surely provide enough accuracy, but at the price of unacceptable computational (and model) complexity. The paper identifies the key phenomena such as preamble detection, central-frequency misalignment, channel captures, vehicles relative speed, that represent the ‘state’ of the MDP modeling the transmission chain, and propose an MDP structure to exploit it. The focus is on 802.11p and OFDM-based PHY layers, but the model is extensible to other transmission techniques easily. The design is tailored for implementation in ns-3, albeit the modeling principle is general and suitable for every event-driven simulator.

**Index Terms**—PHY layer simulation; VANET; vehicular networks; ns-3 simulation; Markov Decision Process; Stochastic models.

## I. INTRODUCTION, MOTIVATION AND GOALS

Simulations are an integral part of networking research. They can provide results about any kind of communication, from wired to wireless, and about any kind of network level, from application down to physical. Moreover, results can be obtained quickly and cheaply, helping the researchers in the fast improvement of their work. Experiments using real equipment and devices are expensive, difficult to perform due to the near impossibility of controlling the environment (think to interferences in wireless communications), and their results are often difficult to reproduce.

On the other hand, simulations can lack in realism, and so they can result in a poor scientific value. The value of a simulation model lies in its ability to capture the relevant phenomena with an affordable computational effort. Traditionally, packet network simulators have relied on very simple transmission models (if any) based on an appropriate distribution function

yielding the probability of successful packet reception. In mobile networks, and in vehicular networks in particular, this simple approach fails due to the tight intertwining of PHY, MAC and routing function. In cooperative driving scenarios, even the application itself can be tightly coupled to the PHY layer, so that reliable application level results require accurate PHY simulations. So in the field of Vehicular Ad-Hoc Networks, the realism of the PHY layer is one of the main concerns.

The problem of accuracy of wireless simulations is highlighted in different papers. Chen et. al. [1] focuses on the problems of the ns-2 network simulator<sup>1</sup>, and introduces an improved version of the 802.11 PHY and MAC layers. Concerning the PHY, the paper develops cumulative SINR computation, a 2-phase reception (i.e., PLCP preamble/header and payload), and physical layer captures. Some aspects, however, are not yet realistic enough: for example, the preamble is detected only if the SINR is over a fixed threshold. A similar policy is applied to the payload. This model has also been ported into ns-3<sup>2</sup>, but it was never included into the official release, so most probably the majority of the researchers still use the default ns-3 implementation, which is more sophisticated than the ns-2 version, but still too coarse to provide reliable results for safety applications. The ns-3 default PHY model (called YANS) [2] uses, instead of a SINR threshold, an analytic formula for Bit Error Rate (BER): given a particular SINR and modulation scheme, the simulator computes the probability of reception and the decision of acceptance/rejection is taken probabilistically<sup>3</sup>. The model, however, does not make any distinction between PLCP preamble/header and payload, and it is not able to model capture phenomena.

Another popular network simulator is OMNet++<sup>4</sup>. Its 802.11 PHY behaves similarly as ns-2 and ns-3. The frame is considered (i.e., preamble is detected) only if the SINR is over a certain threshold. If the preamble is detected, then the probability of reception for PLCP header and MPDU are computed using formulas as in ns-3 and then the acceptance/rejection decision is taken. Again, the frame is considered as a unique

<sup>1</sup><http://isi.edu/nsnam/ns/>

<sup>2</sup><http://www.nsnam.org>

<sup>3</sup>See Sect. II for the details

<sup>4</sup><http://www.omnetpp.org>

“block” and captures are not handled.

To reach a high level of realism, an emulative approach is presented by Papanastasiou et. al. [3]. The authors propose a very detailed PHY and channel model, basically implementing an emulator of 802.11a/p PHY layers for ns-3, coupled with several channel models. This emulator, called PhySim, faithfully reproduces the frame construction and signal processing procedures described in the IEEE 802.11 standard [4] for OFDM-based communications, such as OFDM modulation, interleaving, convolutional encoding and decoding, preamble detection, etc. The signal at the transmitter is sampled based on Nyquist theorem with a proper number of samples per symbol (80 as default). The channel model is sample-based and represented by a tapped delay line (in practice a FIR filter) which can model with detail also frequency and time selective fading, provided that the number of taps and the taps coefficients are known.

This “DSP-oriented” approach naturally reproduces realistic phenomena; however the price paid in computational complexity is humongous. As outlined by Mittag et. al. [5], the computational effort increases by a factor which ranges from a minimum of 300 (for path-loss only experiments) up to a maximum of roughly 14,000 (for experiments considering fast fading) compared to the standard ns-3 model. A significant speed-up (i.e., a factor of 70) is obtained by employing GPU-based computing [6], but such speedup is technology and not model based: employing it with an efficient stochastic model would empower exploring scenarios with thousands of nodes. Moreover, computing clusters normally available to researchers are CPU (not GPU) based, and such slowdown factors make it impossible to perform any experiment encompassing the transmission of more than a few hundred (maybe thousand) packets with a few cars, jeopardizing any attempt to explore large-scale mobile networks or vehicular applications as the ones we explored in [7].

Regarding stochastic models, issues like preamble detection and captures have already been highlighted and faced. Nevertheless, a modern simulator like ns-3 still disregards these aspects. Furthermore, new concerns for VANET simulations arise, which are *i*) time and frequency selective fading, and *ii*) proper shadowing models. Regarding fading, current BER/PER models do not consider the relative speed between sender and receiver. For example, the Nakagami  $m$ -fading model [8], just computes a random attenuation or amplification of the received signal power. Studies performed with PhySim, show instead that the relative speed has a huge impact on the probability of frame reception [5].

Concerning shadowing, the approach is similar as for fading. When a frame is received, a random attenuation or amplification (usually using a log-normal distribution [9]) is applied in order to account for obstacles: obstacles are indeed represented as random objects, an approach that is far from satisfactory when the question is whether an alarm message is received beyond a big truck or behind a blind turn! This aspect is crucial for the evaluation of VANET safety applications. Indeed, literature contains several studies [10]–[13], which,

however, focus either on effects of vehicles or of buildings. These concepts must be merged into a single model in order to perform a correct and comprehensive simulation, even if this means embedding also a scenery description in the simulator. This feat seems indeed far less challenging than embedding a mobility and cooperative driving model, which has already been done, so that vehicle “obstacles” are already available in the simulator, and we just need to take them into account.

As a final remark, a comparison between YANS and PhySim shows that the default ns-3 frame reception rate is much higher than what estimated by PhySim [5]. This aspect was partially due to optimistic BER curves for OFDM, as highlighted by a NIST study<sup>5</sup> and already included in ns-3. However, even with “perfect” BER curves, models like YANS do not have enough state information to apply them correctly, so that first of all we need to identify the state information that is necessary for a correct modeling and then understand how to use it correctly in the model.

“Computational efficiency” and “realism” are the keys of this work, whose aim is to develop an entirely novel physical and channel model based on a Markov Decision Process (MDP) whose state captures enough information of the physical world to enable a correct stochastic representation of the frame receiving procedure, without incurring in the overhead of sample-per-sample DSP-like processing. Our final goal is not substituting PhySim, which can be used, for instance, to explore different algorithms for better frame manipulation, where an MDP model would obviously not be applicable, but to provide the community a tool bridging the gap between PHY emulation and models without enough realism to capture the interaction between the upper layers and the physical realm.

In the remaining part of the paper we use PhySim as reference model for the following two reasons: *i*) performing tests with real hardware is beyond our (present) means and *ii*) in initial experiments, real hardware would not provide evidence on the reasons of a specific result, and understanding what caused an event from the experiments is really difficult. For example, we are interested in knowing if a frame is dropped at the preamble (i.e., a preamble detection error occurred) and the corresponding SINR and relative vehicles speed; or if it is dropped due to a channel capture phenomenon; or what is the impact of the difference between the central frequencies of transmitter and receiver; etc.

The simulator we use is ns-3, since it is widely employed in research and since both the YANS and PhySim models are implemented for it, but the stochastic model we propose can be implemented in any other network simulator.

The remaining part of the paper is organized as follows: Sect. II and Sect. III describe in details the YANS and PhySim models respectively; Sect. IV presents the results of some preliminary tests we performed; Sect. V describes our model proposal and Sect. VI concludes the work.

<sup>5</sup><http://www.nsnam.org/~pei/80211ofdm.pdf>

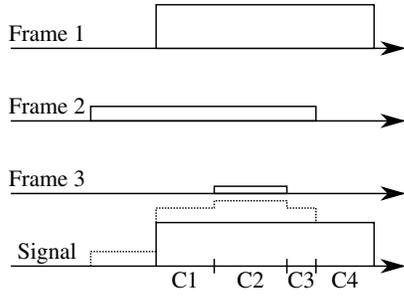


Fig. 1. The chunk-based reception mechanism of YANS. Frame 1 is the frame over the detection threshold, while 2 and 3 are the interfering frames

## II. YANS 802.11 MODEL

This section describes in details how the YANS model of ns-3 works, since it is the model that we use as a reference for stochastic-based reception handling.

The wireless communication model implemented in ns-3 is simple yet effective. When the physical layer sends a frame, it is given to the class implementing the wireless channel. The channel, for every device created in the simulator, computes the propagation delay and the received signal power and communicates to the physical layers of the receivers that a frame is incoming at a certain time and with a certain power. Each station then decides whether the frame will be received or not, based on current state (e.g., a station which is transmitting a frame cannot receive another one in the meanwhile) and on the amount of interference.

The frames' reception model included in ns-3 is based on concept of "chunk", i.e., a received 802.11 frame is divided in into parts which have a constant SINR and bitrate value. Variations of the SINR within a single frame are due to interferences given by overlapping frames. A variation of the bitrate, occur at the end of the PLCP header, when the bitrate speed changes from a standard-defined value to the bitrate chosen for the payload. A simple example (considering only SINR variations) is shown in Fig. 1; the SINR for frame 1 changes four times, so four chunks are considered (C1 to C4). The decision of acceptance (or rejection) is taken probabilistically, i.e., if  $P_e(c_i)$  is the probability of having an error in chunk  $c_i$  (which depends on SINR, bitrate, and bandwidth), then the probability of correctly receive a frame  $f$  is defined as

$$P_r(f) = \prod_{c_i \in f} 1 - P_e(c_i).$$

The details of how  $P_e(c_i)$  is computed, are explained in the original paper [2].

As stated before, Papanastasiou et. al. [3] found this way of computing the reception probability "optimistic", i.e., frames are always correctly received even for low values of SINR (e.g., 1-2 dB for a 6 Mbps transmission). Recently, the error model (i.e., the BER curves) has been updated with an improved version. Still, this model performs differently from PhySim, and is not able to capture effects due to time and

frequency selective fading. This is not, however, the only issue. Another problem is the way in which receptions are handled through the simulator state machine. As previously stated, the decision whether a frame can be received or not is based not only on SINR, but also on the current state. In particular, when a station begins to receive a frame over the detection threshold, if it is in the *IDLE* state then the reception phase starts and the receivers switches to the *RX* state. While in the *RX* state, new incoming frames are only treated as noise and cannot be received. In wireless networks, and in particular in VANETs, situations like the one depicted in Fig. 2(a), due to hidden terminal and similar phenomena, are common. If a station which is far from a receiver sends a frame, and shortly afterwards a much closer node does the same, the receiver will simply try (without success) to get the first and treat the second as noise. However, as shown by Lee et. al. [14], real wireless devices are able to handle such situations and "hook" on the strongest signal, resulting in a correct frame decoding. This phenomenon is known as "capture".

Moreover, YANS treats a frame as a unique block, so a receiver remains always in the *RX* state for a time equal to frame's duration. But the transition to the *RX* state should be performed after *i*) preamble detection and *ii*) after header decoding. If the preamble is not detected, then the reception is not performed. The same happens if the preamble is detected, but the parity check of the PLCP header fails. So the problem depicted in Fig. 2(a) could be partially solved by considering preamble and header as shown in Fig. 2(b). If the simulator considers the failure of the preamble detection phase of frame 1, the receiver can return in the *IDLE* (or *CCA\_BUSY*, depending on energy level) state and correctly process and receive frame 2. This is, however, only a partial solution which would not work in the case of the overlapping of two (or more) preambles. This highlights the intrinsic limitations of

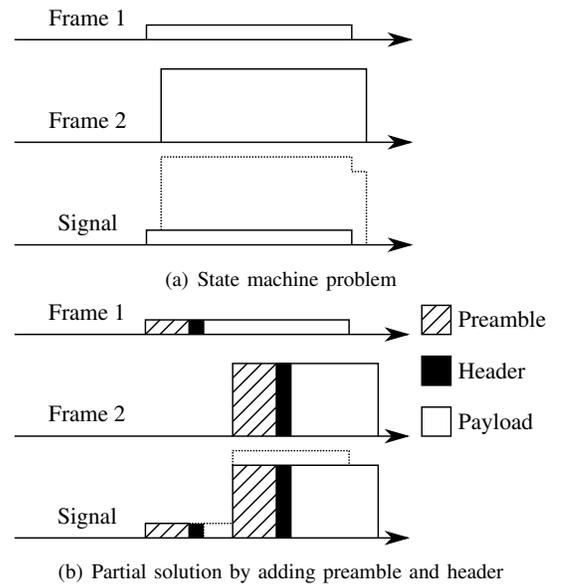


Fig. 2. Incorrect frame handling of the YANS PHY model and partial solution

YANS, and the need of an improvement and the development of a more realistic physical layer, together with the already mentioned issues of embedding more sophisticated models of fading and shadowing taking into account the environment and the relative speed of vehicles.

### III. 802.11 PHYSICAL LAYER AND PHYSIM

As already mentioned, Papanastasiou et. al. [3] have developed a very sophisticated and realistic physical layer model for OFDM-based wireless communications. This model, called PhySim, is basically a “software implementation” of a real wireless card, in the sense that all operations such as scrambling, convolutional encoding, interleaving, modulation, etc., are performed, so the ns-3 simulator embeds in practice an emulator of NICs. We describe this set of operations to gain insight on how the emulator (and a real 802.11a/p device) works. Since we consider OFDM-based communications, the reader can refer to the chapter 17 of the 802.11 standard for further details [4].

In a real wireless card, the physical layer, after receiving the data bits from the MAC, must construct the frame, which is composed by three main “blocks”: *i*) the preamble, *ii*) the header, and *iii*) the actual payload. The preamble is the same for every frame to send, while the bits of the header and the payload are processed as mandated by the standard. In the first step, data bits are sent into a scrambler, which transforms the sequence of ones and zeros in order to avoid unwanted correlations in the final signal. The following step is convolutional encoding, which adds an error-correcting code depending on the selected data rate<sup>6</sup>. Then, the bits are interleaved, i.e., rearranged in order to give the error-correcting code a higher chance to retrieve the original message in the case of transmission errors. Next, the bits are modulated using either BPSK, QPSK, 16-QAM or 64-QAM, depending on the selected data rate. This operation produces a set of complex numbers representing constellation points, which are grouped into blocks of 48 elements; each element is mapped onto a different OFDM subcarrier. Four pilot subcarriers are added to the set of 48 complex numbers, and the resulting vector is the representation of the OFDM signal in the frequency domain. An Inverse Fast Fourier Transform (IFFT) finally generates the sampled signal to be transmitted. The samples are converted into an analog signal, which is then translated in frequency in the desired band (e.g., 5 GHz for 802.11a) and sent to the antenna. The PhySim emulator performs all these steps, except for the analog and frequency conversions; the signal is represented by a vector of complex samples.

On the receiver side, the emulator performs the signal degradation (i.e., path loss, fading, central frequencies misalignment, etc.) by using a channel represented by a tapped delay line. Then it begins the reception procedure by trying to detect the preamble and synchronize on the clock of the sender. If the preamble is correctly detected, then the receiver

<sup>6</sup>For example, for the 6 Mbps data rate (20 MHz channel), a code rate of 1/2 is used, which means that for every data bit, a bit for the convolutional code is added. See Tab. 17.3 in the standard.

tries to decode the header and finally the payload by using the inverse transmission procedure.

### IV. PRELIMINARY ANALYSIS

The tests presented here are not meant for deriving the probability distributions needed for the final model: their goal is giving a better understanding of the dynamics of a real 802.11 physical layer and identify the state variables needed for the MDP model.

Our analysis begins with two basic tests; *i*) one transmitter and a set of 30 receivers randomly placed in a stretch of highway of 2 km without any interference and *ii*) one transmitter and one receiver plus a node which generates interference in a controlled way. For simplicity, both preliminary tests consider a channel with only path-loss. We employ 802.11p CCH, so we consider physical layer parameters of a 10 MHz channel.

#### A. Single transmitter test

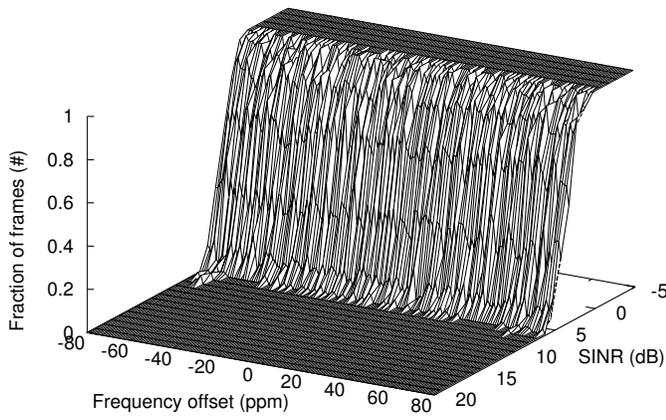
The aim of this test is to analyze the behavior of the physical layer when only floor noise is present. We consider a transmission speed of 6 Mbps and an MPDU of 200 bytes. Fig. 3 shows what happens to frames for different values of SINR and frequency offset. The allowed frequency offset for 802.11p is 20ppm: this means that sender and receiver can be at most 40ppm misaligned (if the card behaves as mandated). Someone could, however, be interested in analyzing, for example, what happens if wireless cards are faulty. With PhySim this is possible, so we intentionally extend our analysis up to 80 ppm.

When SINR approaches 0 dB, the emulator, regardless of the frequency offset, is not able to detect the preamble. As the SINR increases, the card can easily detect the preamble and the frequency offset does not play a significant role if the offset is within the requirements of the standard. When operating outside mandated range, the card is able to detect the preamble, but seems to wrongly estimate the offset, since frames are dropped either at PLCP header or at payload, regardless of the SINR.

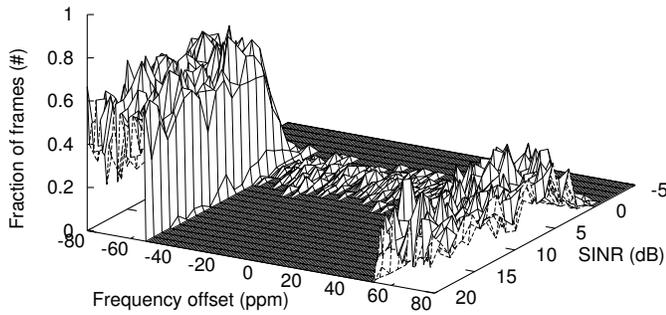
Fig. 4 instead shows the events at the physical layer, without considering the frequency offset which, in this experiment, is set as mandated by the standard (i.e.,  $\pm 20$ ppm per wireless card). The figure shows that the approach used in ns-2 and in OMNet++ for preamble detection (i.e., threshold based) is a rough approximation of what happens in reality. For example, for a SINR of 3-4 dB, 30 to 60% of the preambles are detected and the remaining are not. Then, we see that, if the preamble is correctly detected, the number of frames discarded at header is marginal (remember that we are considering only noise and not interference), and that payload drops occur, with different probabilities, up to 10 dB SINR.

#### B. A Simple Interference Test

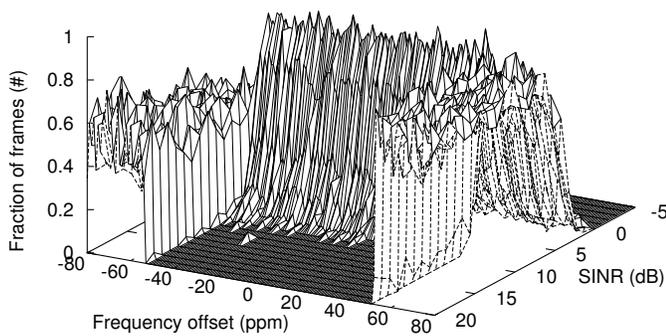
In this test we fix the position of the receiver while moving the sender and an interfering node. Fig. 5 shows the scenario. We employ two moving cars: the first one (S in the figure) which is the sender of the frame we want to analyze, while



(a) Preamble drop



(b) Header drop



(c) Payload drop

Fig. 3. Fraction of frames dropped as a function of the SINR and the frequency offset, for the experiment with no interference. The data rate is 6Mbps with an MPDU size of 200 bytes

the second (I) acts as interfering node. When the sender node sends a frame, the interfering node does the same, so that the packet arrives at the receiver (R) delayed (with respect to the one sent by the sender) by a desired amount of time<sup>7</sup>. The receiver node R, monitors what happens to the frame sent by S (i.e., if it can be detected, decoded and received). We arbitrarily change the distances  $D_i$  and  $D_s$ , so we obtain different amount of interference caused by the different positions of sender and interferer. We use delay values of 100, 4000, 8000, 16000, and 24000 ns.

Fig. 6 plots the fraction of frames sent by S and dropped at

<sup>7</sup>We compensate propagation delay, so the interfering frame arrives at the exact desired delay.

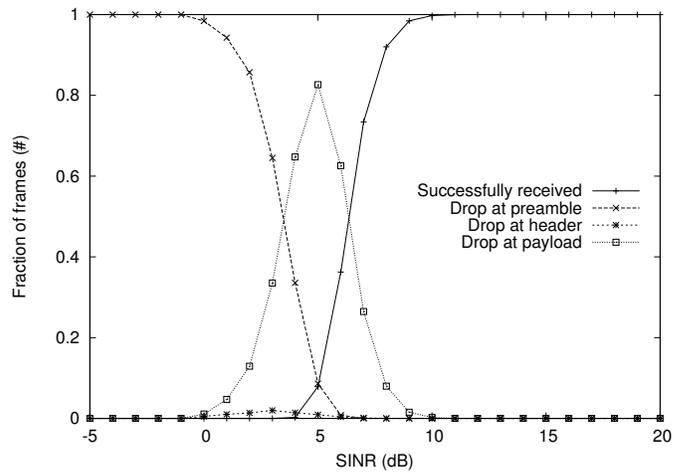


Fig. 4. Analysis of PHY layer events as a function of the SINR, for the experiment with no interference. The data rate used is 6Mbps with an MPDU size of 200 bytes

preamble as a function of the SINR. Notice that the SINR is computed only when the two frames overlap. So the SINR in the figure gives an idea of the amount of disturbance during the interfered part of the preamble: the lower the SINR, the higher the interference. A simple model like those in Omnet++ and ns-2 would simply draw a straight line at some given SINR: if below the frame is not decoded, if above decoding is attained with probability 1. YANS instead would compute the average SINR over Preamble and PLCP Header and compute a cumulative BER: a more sophisticated approach, but just

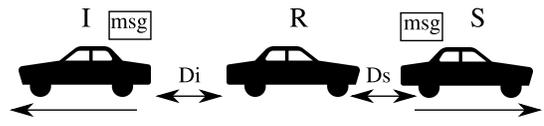


Fig. 5. Scenario of the simple interference test

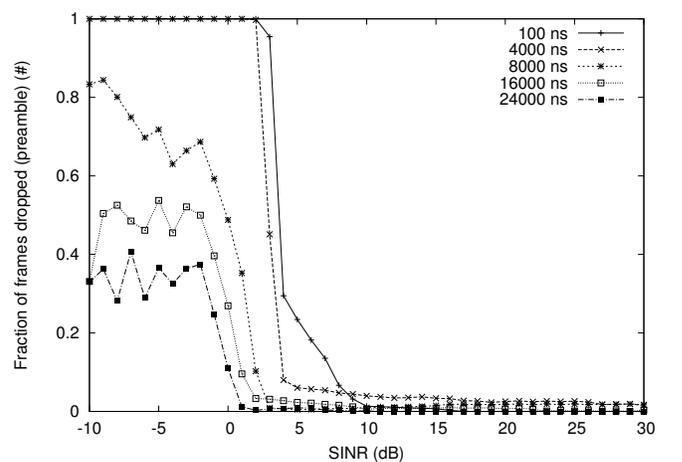


Fig. 6. Fraction of frames dropped at preamble as a function of the payload SINR for frames from S in the experiments with interference. The data rate used is 6Mbps with an MPDU size of 200 bytes

as far from reality. PhySim instead implements the correlators that in NICs actually perform preamble synchronization, which is a different procedure from demodulation and decoding of symbols. Also PhySim however introduces some modeling approximations. It computes the symbol-by-symbol SINR on the entire preamble (four symbols), computes the average of these values and attempts detection only if this average is above 4dB (the value can be changed).

As we see from Fig. 6, this model leads to very different behaviors when the frame from car I starts almost together with the frame from car S (100 ns) or when it is delayed. The  $x$  axis is the SINR at payload, and not the SINR averaged over the preamble as PhySim does. We cannot state if the model of PhySim is correct or not, but we notice that temporal offsets of interferent signals can play a significant role in frame detection and even more on capture phenomena: a stronger signal starting soon after a weak one can lead to a successful capture, while one starting later on, even if the strength is the same, may not be worth the change to the new frame, since the detection probability of the first one remains high.

## V. PROPOSED MODEL

The results in Sect. IV highlight a stiff scenario: event-based, stochastic PHY models fast enough to empower application-level analysis are not accurate enough, while detailed PHY simulations yielding enough accuracy are too slow to allow any meaningful insight in applications. Networking and applications simulations are best designed in an stochastic event-driven architecture (as the one of ns-3), while traditionally transmission systems and PHY layer simulations have exploited a time-driven approach, with the capability of changing the simulation domain from time to frequency as needed, an approach enabled by a time-driven sampling of the analog phenomena, which is prevented in a purely event-driven simulation, where the system is not really “sampled” but it is actually a discrete-time system whose state outside the discrete-time instants is simply undefined.

Keeping everything within a single simulator is extremely difficult, CPU consuming, and also dangerous, since the simulator will end up having a complexity and a number of parameters so large to jeopardize the sheer understanding of its results. If the goal is the analysis and the design of better protocols and systems, and not the analysis and design of better coding schemes, modulations or signal processors, then the details and complexities of a “software implementation” of the transmission chain are not needed. What is needed is an accurate enough stochastic, event-driven model that correctly represent meaningful situations in terms of networking.

A more accurate analysis of the problem suggests that the lack of accuracy of event-based models derives more from a lack of information driving the model than from conceptual limitations of an event-based stochastic model itself, thus our research path is in the direction of defining a better state-based stochastic model.

Formally, what we are suggesting, is the definition of an event-based PHY level model, where the event end-of-frame

at a receiver, starts the Monte Carlo solution of a discrete time Markov Decision Process (MDP), where the probability of correctly receiving a frame is the outcome of a walk onto the MDP state space  $\mathcal{S}$ . As already mentioned, an attempt in this direction, implemented in ns-2, but never included in the standard distribution of ns-3, was suggested in [1], but it was based more on the observation of how real receivers work than on the identification of the sufficient state information for a model to correctly represent the channel. We notice incidentally that also the YANS model is implicitly an MDP, since the correct decoding of the packet is “declared” if and only if all the chunks of the packet, each with a different interferent level, are declared correctly decoded. In practice the interferent level, and the interferent level only, is used as state of an MDP, where the decoding model in each state is represented by a random variable conditioned on the SINR. The RV can take into account the attenuation only or attenuation plus a fading model. In a complex scenario like a VANET, however, disregarding the vehicles speed, the *structure* of the interferent, and finally the structure of the packet being received, is too simplistic, and as discussed in Sect.IV, leads to unrealistic results.

The walk in our model always starts from a state  $\mathcal{S} = F_S$ , which identify the attempt of the receivers correlators to synchronize a preamble, and will always terminate in one of two absorbing states  $\mathcal{S} = F_R; F_D$ , representing the correct Reception or the Discard of the frame respectively.

First of all we observe that each frame reception phase (preamble, header and payload) has different characteristics, so they need a different probabilistic modeling of correct reception. Let  $R_P = \{\text{Preamble, Header, payLoad}\}$  be the state variable describing the receiving phase. Next, channel capture phenomena need to be taken into account, specially in VANETs, because in dense traffic scenarios one can expect that a receiver is nearly continuously receiving signals above the sensitivity threshold, and a ‘smart’ receiver will consistently try to decode the packet with the highest probability of success, so that, when a new ‘interferent’ is heard with power much higher than the signal presently being received, the receiver will try to ‘jump’ onto this new signal<sup>8</sup>.

Next, we observe that considering only the overall power of the interfering signals is a poor description of what happens in real channels: the summation of tens of negligible contributions can be treated as additional noise, while a single high power interferent has a very different impact, and may lead to channel captures. Thus we identify, as another variable describing  $\mathcal{S}$ , the vector of interfering signals  $\vec{I}$  with all their characteristics, and not only the SINR as used in simple packet-level models. This means that we maintain all the information related to all frames on-air. At first sight this seems a tremendous effort, but indeed this information is always present in simulators, but it is normally used only to compute the SINR. The reason lies probably in the conceptual

<sup>8</sup>Although this may seem very difficult to implement, this function is already implemented in some 802.11 chipsets, since the event is marked by a sudden increase of the received power, which is easy to detect.

additional complexity of a multivariate state.

Each frame can be described with a tuple  $\mathcal{F}$  representing its characteristics. In particular,  $\mathcal{F}$  includes a basic representation of the frame, i.e., start ( $t_s$ ) and end ( $t_e$ ) time, signal power  $PW$  and number of data bits  $B$ . These are all the parameters normally taken into account in packet-level simulators, but we have seen in Sect. IV that many others may be at the heart of differences between detailed PhysSim results and YANS ones. The frequency offset  $\Delta_f$  between the clocks of the transmitter and the receiver before synchronization, plays a fundamental role in the decoding success, since it is the key parameter driving the preamble detection phase. The modulation and coding scheme used for the transmission influence the correct frame reception. We can add the variable  $MC$  to the state, which takes values in the combination of modulation and coding admitted by the protocol. Finally, the difference in speed  $\Delta_v$  between sender and receiver is fundamental for considering fading effects, and it is important for every frame, not only for the one under detection. To summarize, each frame (under reception or interfering) is described by

$$\mathcal{F} = (t_s, t_e, PW, B, \Delta_f, MC, \Delta_v).$$

The interference  $\vec{I}$  is a set of such descriptors  $\mathcal{F}$ . We can visualize the instantaneous interference at time  $t$  as the set of  $\mathcal{F}$  for which  $t_s < t < t_e$ .

In order to properly consider shadowing phenomena, we need a description of the environment  $E$ . Such description enables the possibility to compute the attenuation caused by occluding objects, like other vehicles or buildings.  $E$  can be considered constant during the reception of a particular frame, as objects movements in less than 6 ms (maximum MPDU at minimum speed) is negligible. On the other hand,  $E$  varies in both space and time depending on the relative position of sources on the receiver.

The state of the MDP describing the reception of a frame is thus formally and fully described as

$$\mathcal{S} = \{F_S; F_R; F_D; (R_P, \vec{I}); E\}.$$

The MDP state is discrete, but infinite because  $\vec{I}$  can assume infinite values. While this can be a stiff problem when the goal is the analytic solution of the MDP, it is not an issue for the implementation of the model in a simulator and its point solution with a Monte Carlo walk on the state space. In any case, if required, a proper quantization can make the state space finite with measurable approximation. Transitions in the MDP are triggered by changes in  $\mathcal{S}$ , which are the conditioning random variables driving the model. The transition probabilities

$$P[S_j | S_i]$$

can be either model-driven or trace-driven. In a model-driven approach, they are defined by the communication link model adopted. This is in practice the residual FER (Frame Error Rate) given the raw BER as a function of thermal noise, distance loss, shadowing, fading, interleaving and error correcting

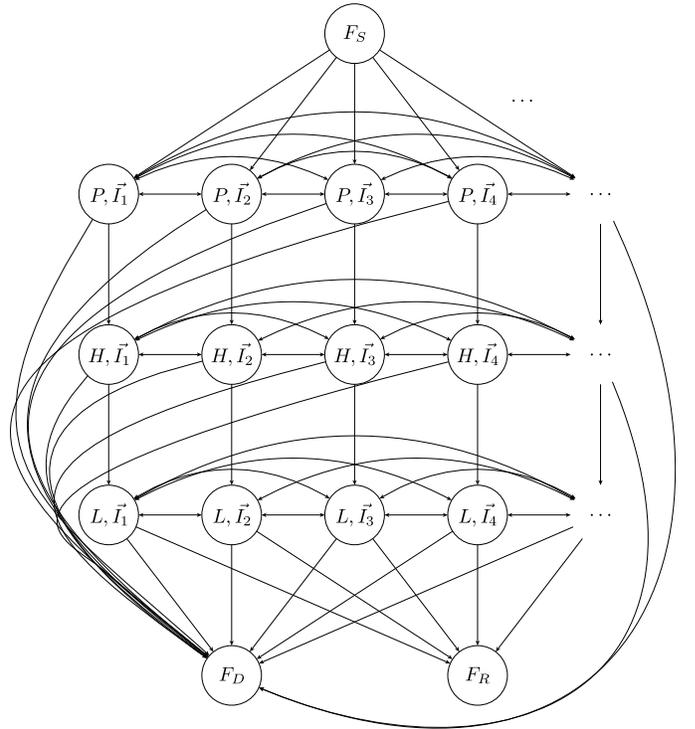


Fig. 7. State diagram of the MDP for the model

code<sup>9</sup>. In a trace-driven approach instead the residual FER is simply derived in tabular form from a data-base of measures. This second approach has the advantage of conceptual simplicity and the robustness of experimental science, but requires a complex and expensive measurement campaign, which, to the best of our knowledge, has never been undertaken for VANETs. Indeed, such a campaign, would be badly needed also for validating and tuning communication link models, but this issue seems to be vastly ignored by both academia and industry.

To better explain the idea of the model, Fig. 7 shows the states of the MDP. Starting from  $F_S$  the walk on the MDP moves to a state with  $R_P = P$  (preamble detection) and the proper interference vector  $\vec{I}_i$ . If the interference changes (e.g., a new overlapped frame begins), then the walk moves to another state  $\mathcal{S}_j = \{P, \vec{I}_j\}$  corresponding to the new interference  $\vec{I}_j$  or directly to  $F_D$  if the model in  $\mathcal{S}_i$  stated that the preamble cannot be detected. When the preamble terminates, the receiver can move either to the header decoding phase ( $R_P = H$ ) or once more to the absorbing state  $F_D$ , indicating that the preamble was not successfully detected. Also in case the new interference  $\vec{I}_j$  indicates a capture, transition is directly to  $F_D$ . For header ( $H$ ) and payload ( $L$ ) the procedure is similar. At the end of payload decoding, the frame can either be received ( $F_R$ ) or discarded ( $F_D$ ).

For both model correctness and computational efficiency,

<sup>9</sup>We stress once more that we propose the use of a stochastic model conditioned model, and not the software implementation of the symbol-by-symbol detection process as done in PhysSim.

when the walk finally reach one of the absorbing states, the simulator can set a flag on all the frame reception events (corresponding to other walks on the MDP) that have become impossible, for instance for all frames whose starting times  $t_s$  are between the starting time of the frame under detection and the time instance when the walk terminated.

For what computational complexity is concerned, we deem that this model is only marginally more complex than YANS, since the walk on the MDP has a negligible cost, so that the potential additional cost is only related to more complex transmission-link models that can be included, or to the execution of a larger number of them due to the richer state descriptors that force more transitions in the MDP (remember that also YANS is based on an MDP even if not expressly designed as such).

## VI. FINAL DISCUSSION

Simulation is becoming the technique of choice for the evaluation of complex systems, where an experimental approach is too expensive and analytical modeling too complex or simply not available.

All too frequently, however, the models underlying simulations are not fully understood or are taken “as such” by researchers, without a proper design process, or without a sane application of the Occam’s Razor.

The case of mixing together the simulation of the transmission chain with packet networks and applications is one of the toughest ‘environments’ for finding the appropriate simulation modeling abstraction level. This tough case, however, needs to be tackled at least for safety applications in cooperative driving, where the outcome of the application is so much intertwined with the operations of the physical layer that entirely decoupling them, as in most simulators, is simply unacceptable.

This work has discussed limitations of the most widespread PHY simulators and the different outcomes that are obtained when the PHY layer is actually emulated within the simulator using a DSP-like approach.

Finally, as probably the main contribution of the work, it has been discussed that the limitation of current models does not lie in the event-driven approach, but in not considering enough state information to take a correct (stochastic) decision on the frame decoding probability. This limitation can be overcome by using a model based on a Markov Decision Process, whose state contains enough information to correctly condition the stochastic decision on whether the frame must be dropped or its decoding should be continued.

We have started the implementation of this framework within ns-3, but results are not yet available, specially because the identification of the proper conditioned random variables that are associated to the MDP states requires an extensive study to identify if proper models exist in the communication theory literature and, if they do not exist either extensive measurement campaigns, or simulations with tools like MatLab or the same PhySim emulator used in this preliminary study.

## ACKNOWLEDGMENT

We would like to thank Jens Mittag of the Karlsruhe Institute of Technology for his help in understanding PhySim, the useful discussions and for constantly providing us with the latest release of the emulator.

The Autonomous Province of Trento (PAT) partially funded the research that led to this paper.

## REFERENCES

- [1] 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, MSWiM '07*, 22-26 Oct. 2007.
- [2] M. Lacage and T. R. Henderson, “Yet another network simulator,” in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, ser. WNS2 '06, 10 Oct. 2006.
- [3] 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*, 18-21 Apr. 2010.
- [4] “IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, Jun. 2007.
- [5] J. Mittag, S. Papanastasiou, H. Hartenstein, and E. G. Ström, “Enabling accurate cross-layer phy/mac/net simulation studies of vehicular communication networks,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1311–1326, Jul. 2011.
- [6] P. Andelfinger, J. Mittag, and H. Hartenstein, “GPU-based architectures and their benefit for accurate and efficient wireless network simulations,” in *Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*, 25-27 Jul. 2011.
- [7] M. Segata and R. Lo Cigno, “Emergency braking: A study of network and application performance,” in *Proceedings of the eight ACM international workshop on VehiculAr InterNetworking, VANET '11*, 23 Sep. 2011.
- [8] M. Nakagami, “The  $m$ -distribution, a general formula for intensity distribution of rapid fading,” in *Statistical Methods in Radio Wave Propagation*, W. G. Hoffman, Ed. Oxford, England: Pergamon, 1960.
- [9] H. Hartenstein and K. Laberteaux, Eds., *VANET Vehicular Applications and Inter-Networking Technologies*. John Wiley & Sons, Jan. 2010.
- [10] E. Giordano, R. Frank, G. Pau, and M. Gerla, “Corner: a step towards realistic simulations for vanet,” in *Proceedings of the seventh ACM international workshop on VehiculAr InterNetworking, VANET '10*, 24 Sep. 2010.
- [11] M. Boban, T. Vinhoza, M. Ferreira, J. Barros, and O. Tonguz, “Impact of vehicles as obstacles in vehicular ad hoc networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 1, pp. 15–28, Jan. 2011.
- [12] R. Meireles, M. Boban, P. Steenkiste, O. Tonguz, and J. Barros, “Experimental study on the impact of vehicular obstructions in vanets,” in *Vehicular Networking Conference (VNC), 2010 IEEE*, 13-15 Dec. 2010.
- [13] C. Sommer, D. Eckhoff, R. German, and F. Dressler, “A computationally inexpensive empirical model of ieee 802.11p radio shadowing in urban environments,” in *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*, 26-28 Jan. 2011.
- [14] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi, “An experimental study on the capture effect in 802.11a networks,” in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization, WinTECH '07*, 10 Sep. 2007.